Applications of Random Graphs to Design and Analysis of LDPC Codes and Sensor Networks

A Thesis
Presented to
The Academic Faculty

by

Hossein Pishro-Nik

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2005

Applications of Random Graphs to Design and Analysis of LDPC Codes and Sensor Networks

Approved by:

Dr. Faramarz Fekri, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ali Adibi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ian F. Akyildiz
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christopher E. Heil
School of Mathematics
Georgia Institute of Technology

Dr. Steven W. McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: August 18, 2005

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis investigates a graph and information theoretic approach to design and analysis of low-density parity-check (LDPC) codes and wireless networks. In this work, both LDPC codes and wireless networks are considered as random graphs. This work proposes solutions to important theoretic and practical open problems in LDPC coding, and for the first time introduces a framework for analysis of finite wireless networks.

LDPC codes are considered to be one of the best classes of error-correcting codes. In this thesis, several problems in this area are studied. First, an improved decoding algorithm for LDPC codes is introduced. Compared to the standard iterative decoding, the proposed decoding algorithm can result in several orders of magnitude lower bit error rates, while having almost the same complexity. Second, this work presents a variety of bounds on the achievable performance of different LDPC coding scenarios. Third, it studies rate-compatible LDPC codes and provides fundamental properties of these codes. It also shows guidelines for optimal design of rate-compatible codes. Finally, it studies non-uniform and unequal error protection using LDPC codes and explores their applications to data storage systems and communication networks. It presents a new error-control scheme for volume holographic memory (VHM) systems and shows that the new method can increase the storage capacity by more than fifty percent compared to previous schemes.

This work also investigates the application of random graphs to the design and analysis of wireless ad hoc and sensor networks. It introduces a framework for analysis of finite wireless networks. Such framework was lacking from the literature. Using the framework, different network properties such as capacity, connectivity, coverage, and routing and security algorithms are studied. Finally, connectivity properties of large-scale sensor networks are investigated. It is shown how unreliability of sensors, link failures, and non-uniform distribution of nodes affect the connectivity of sensor networks.

CHAPTER I

INTRODUCTION

This work revolves around three main areas, error-control coding, random graph theory, and sensor and ad hoc networking. It uses random graph theory as a tool in the study of both error-correcting codes and wireless networks. In error-control coding the focus is on low-density parity-check (LDPC) codes.

Today, error-control codes are a critical part of wireless phones, compact disks and digital versatile discs (CDs/DVDs), satellite communication, paging systems, and hard drives. In a digital communication system, for example, the goal is to transport information bits from one party (sender) to another one (receiver). However, the information transmission is always affected by different sources of noise and interference. Consequently, some of the bits are changed during the transmission. By adding redundant bits to the information bits, error-correcting codes are used to detect and correct transmission errors.

Coding theory began in the late 40's with the work of Hamming and Shannon at Bell-laboratories. Hamming showed how to construct and analyze the first practical error control systems; while Shannon found the limits for ideal error control. The main idea behind error-control coding is to add redundancy to the information before the transmission, and use this redundancy at the receiver to correct the errors.

The emergence of graph-based error correcting codes in the last decade has revolutionized the area of error-control coding. One of the most promising classes of graph-based codes is the class of low-density parity-check (LDPC) codes. These codes can be decoded using efficient iterative decoding algorithms. Empirical results show that LDPC codes can approach the Shannon theoretic limit provided that they have large block lengths. However, many theoretical and practical challenges in the area need to be answered. The goal of this work is to propose solutions to some of the most important practical and theoretical challenges in this area.

The contributions of this work in error-control coding include:

- We introduce an improved iterative decoding algorithm that significantly reduces the error probability without increasing complexity. The algorithm is specifically useful in practical applications in which the block length cannot be large. In these scenarios, it has been an open problem to design good LDPC codes. Our improved decoding method proposes a solution to this problem. Instead of trying to find better codes, it improves the performance of the existing codes.

- We introduce a framework for design and analysis of LDPC codes for non-uniform and unequal error protection. We show that a variety of applications can be included in this framework. We study LDPC codes for channels with non-uniform noise distributions, rate-adaptive coding, and unequal error protection.

- We propose a new error-control scheme for volume holographic memory (VHM) systems based on LDPC codes. We optimize high-rate LDPC codes for the nonuniform error pattern in holographic memories to reduce the bit error rate (BER) extensively. The prior knowledge of noise distribution is used for designing as well as decoding the LDPC codes. We show these codes have a superior performance to that of Reed-Solomon (RS) Codes. Our simulation shows that we can increase the maximum storage capacity of holographic memories by more than 50 percent if we use the proposed LDPC codes with soft decision decoding instead of conventionally used RS codes with hard decision decoding. The performance of these LDPC codes is close to the information theoretic capacity.

- We study fundamental properties and optimal design of rate-compatible LDPC codes. We first prove that for any ensemble of LDPC codes, there exists a puncturing threshold p^*. If the puncturing fraction p is smaller than p^*, then the punctured code is asymptotically good. On the other hand, if $p > p^*$, error probability is bounded away from zero, independent of the communication channel. We find these puncturing thresholds for both randomly and intentionally punctured LDPC codes. We then

prove that for any rates R_1 and R_2 satisfying $0 < R_1 < R_2 < 1$, there exists an ensemble of LDPC codes with the following property. The ensemble can be punctured from rate R_1 to R_2 resulting in asymptotically good codes for all rates $R_1 \leq R \leq R_2$. Specifically, this implies that rates arbitrarily close to one are achievable via puncturing. We also show that punctured LDPC codes are as good as ordinary LDPC codes. For BEC and arbitrary positive numbers $R_1 < R_2 < 1$, we prove the existence of the sequences of punctured LDPC codes that are capacity achieving for all rates $R_1 \leq R \leq R_2$. Based on the above observation, we then propose a method to design good punctured LDPC codes over a broad range of rates. The method is very simple and does not suffer from the performance degradation at high rates. Finally, we show that these results may be used for the proof of the existence of the capacity-achieving LDPC codes over binary-input output-symmetric memoryless channels.

- Finally, we study the performance limits of LDPC codes under different scenarios. We study performance limit of LDPC codes that have linear minimum distance. These codes are practically important, for example when it is necessary to avoid error floor. We obtain lower and upper bounds on the performance of these codes. We also study performance limit of maximum-likelihood (ML) decoding of LDPC codes over the binary erasure channel (BEC).

The second part of this work considers wireless sensor networks. Sensor networks have received a great deal of interest lately, with proposed applications in military and civilian surveillance and sensing tasks and potential services that would enhance the ability of the growing domain of wireless technologies [4]. Wireless sensor networks have benefited from advances in both MEMS technology and networking. In sensor networks, a large number of sensor nodes are usually randomly deployed in a target area to perform a collaborative sensing task. For example, suppose we want to monitor the temperature in a certain area. The sensors are distributed in the area, and each sensor senses the temperature, and send data to a sink hop by hop. There maybe several sinks in the network. Sensors have been used in our everyday life since long time ago. However, in a sensor network, we have a

network that uses the collaborative effort of a large number of sensors.

In this work we are interested in graph and information theoretic properties of sensor networks. The contributions of this work in sensor networking include:

- We introduce a framework to study finite (practical-sized) wireless sensor and ad hoc networks. In the past, many analytic results for wireless networks have been reported for the case where the number of nodes n in the network tends to infinity (large-scale networks). These include connectivity, coverage, and capacity. These results have not been extended for small or moderate values of n, although in many practical networks n is not very large. We first show that previous asymptotic results provide poor approximations for the finite networks (small-scale networks). We then aim to develop a framework to analytically study network properties without assuming that n is large. We provide a set of differences between small-scale and large-scale analysis. We consider wireless networks in which the location of the nodes is random. We study routing algorithms, coverage, connectivity and capacity of finite wireless networks. We provide easily computable expressions for different network properties. With validation from simulations, we show that these analytic expressions give very good estimates of these quantities for finite wireless networks. Our investigation suggests that the small-scale networks posses unique characteristics that require a new framework for analysis and design.

- We study connectivity properties of large-scale wireless sensor networks. In wireless sensor networks, both nodes and links are prone to failures. We study connectivity properties of large-scale wireless sensor networks and discuss their effect on routing algorithms and network reliability. We assume a network model of n sensors which are distributed randomly over a field based on a given distribution function. The sensors may be unreliable with a probability distribution, which possibly depends on n and the location of sensors. Two active sensor nodes are connected with probability $p_e(n)$ if they are within the communication range of each other. We prove a general result relating unreliable sensor networks to reliable networks. We investigate different

graph theoretic properties of sensor networks such as k-connectivity and the existence of the giant component. While connectivity (i.e, $k = 1$) insures that all nodes can communicate with each other, k-connectivity for $k > 1$ is required for multi-path routing. We analyze the average shortest path of the k paths from a node in the sensing field back to a base station. It was found that the lengths of these multiple paths in a k-connected network are all close to the shortest path. These results have been shown through graph theoretical derivations and also have been verified through simulations.

CHAPTER II

BACKGROUND

2.1 Error Control Coding

Error-control coding is used to reduce the bit error rate (BER) in communication over noisy channels. Block codes are one of the most important classes of error-correcting codes. In block coding, before the transmission, information bits are divided to several blocks. Each block has the same length k. For example in Fig. 1, the block is the vector $(u_1, u_2, ..., u_k)$ where u_i's are bits. Then the information block is mapped to a longer block $(x_1, x_2, ..., x_n)$, which is called the codeword. The length of the codeword, n, is called the code length. Note that $n > k$, thus there is some redundancy in the codeword. In fact, this redundancy is used in the decoding process to detect and correct errors.

Most practical block codes are linear, that is the mapping from the information block to the codeword is a linear mapping. This means that there exists a binary matrix G, the generator matrix, that defines the encoding process. Specifically if $\underline{U} = (u_1, u_2, ..., u_k)$ and $\underline{X} = (x_1, x_2, ..., x_n)$ are the information block and the codeword, respectively, then

$$\underline{X} = \underline{U}G. \tag{1}$$

Equivalently, a linear block code can be defined by a parity-check matrix, H. The parity-check matrix H is a $(n-k) \times n$ matrix satisfying $GH^T = 0$. Equivalently a binary vector $\underline{X} = (x_1, x_2, ..., x_n)$ is a valid codeword if and only if $\underline{X}H^T = 0$.

LDPC codes were first proposed by Galleger [38]. Recently, there has been a tremendous amount of work on these codes, which has resulted in considerable improvement in this area. Mackay revived the interest in LDPC codes and showed that these codes have a lot of good properties [72]. Luby et al. introduced irregular LDPC codes and showed that these codes can provably achieve the capacity of the binary erasure channel (BEC) [71], [70]. Richardson et al. generalized some of these results to a vast class of interesting channels and developed

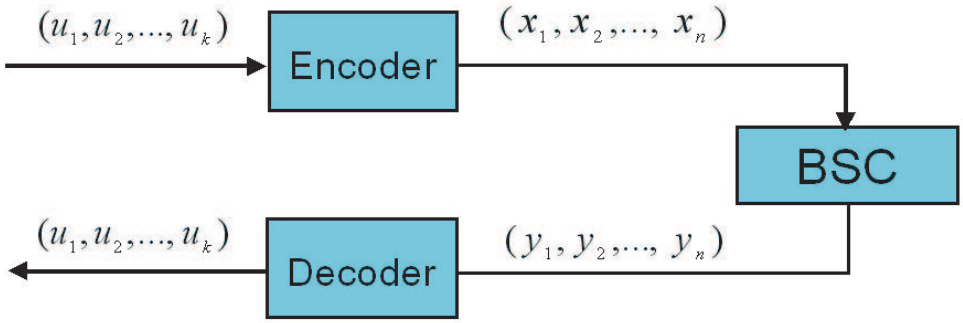

Figure 1: Block codes for noisy channels

a method called density evolution to analyze the iterative decoder of LDPC codes [112], [54]. They used the method to design LDPC codes operating at rates very close to the Shannon limit. These codes were shown to outperform turbo codes [54], [24], [25]. These results encouraged more research on these codes and their applications to practical systems.

An LDPC code is defined as a linear block code with a sparse parity check matrix $H = [h_{ij}]$, i.e., most of the elements of H are equal to 0 and a few of them are equal to 1. For an (n, k) binary linear block code, the parity-check matrix has $m = n - k$ rows and n columns, and codewords $\underline{X}$ are binary vectors of length n that satisfy the equation $\underline{X}H^T = 0$. Each row of H corresponds to a parity-check equation and each column corresponds to one bit of the codewords. An LDPC code can also be represented by a bipartite graph called the Tanner graph [121]. A Tanner graph is a bipartite graph with bipartition V and C, where $V = \{v_1, v_2, ..., v_n\}$ is the set of variable (message) nodes and $C = \{c_1, c_2, ..., c_m\}$ is the set of check nodes. The nodes c_i and v_j are adjacent (connected by an edge) if and only if $h_{ij} = 1$. The degree of a node is defined as the number of edges incident with it. An LDPC code is called regular if the degrees of all message nodes are equal and the degrees of all check nodes are equal. Otherwise the code is called irregular. As an example, Figure 2 shows the Tanner graph of the code defined by

$$H = \begin{pmatrix} 1\,1\,0\,1\,0\,0 \\ 0\,1\,1\,0\,1\,0 \\ 1\,0\,1\,0\,0\,1 \end{pmatrix} \tag{2}$$

It is clear from Figure 2 that the code is irregular because the variable nodes have

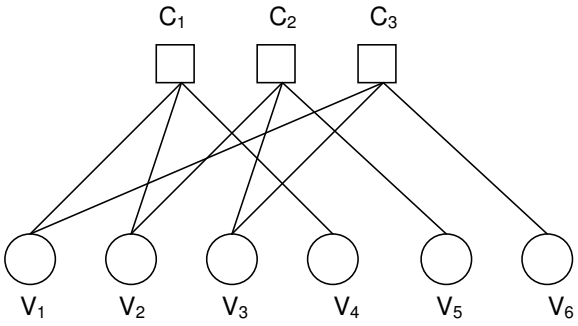

Figure 2: The Tanner graph of an LDPC code

different degrees. The degree distribution of the Tanner graph of an LDPC code is an important parameter. In the rest of the section we review the basic properties of random LDPC codes and present the most important developments in the area. First, we define the ensembles of regular and irregular LDPC codes. We then describe the iterative decoding of these codes and state theorems on the concentration around the average performance and convergence to the cycle-free case. We also describe density evolution and degree optimization.

2.2 Ensembles of LDPC Codes

Here, we define the ensemble described in [112]. The ensemble $\mathcal{C}^n(d_v, d_c)$ of LDPC codes is an ensemble that consists of regular LDPC codes in which variable nodes have degree d_v and check nodes have degree d_c. To construct a graph from the ensemble, we do the following. To each variable or check node we assign d_v or d_c sockets, respectively. We label the variable nodes and check nodes sockets separately with the set $\{1, 2, ..., nd_v\}$. We then pick a random permutation π on $E = nd_v$ letters. For each i, we put an edge between the socket i and $\pi(i)$. Two vertices are connected if there is an edge between their sockets.

For irregular ensembles we need to define the degree distribution. The degree distribution is defined by the pair (λ, ρ), in which λ and ρ are polynomials. In particular,

$$\lambda(x) = \sum \lambda_i x^{i-1}, \quad \rho(x) = \sum \rho_i x^{i-1} \tag{3}$$

where λ_i is the fraction of edges connected to a variable node of degree i and ρ_i is the fraction of edges connected to a check node of degree i. The ensemble $\mathcal{C}^n(\lambda, \rho)$ is defined

similarly to the regular ensembles.We also use $g(\lambda, \rho)$ to show this ensemble. Since LDPC codes are linear block codes, the encoding can be done in $O(n^2)$ time. However, a faster method for encoding is given [113] that in some cases results in linear-time encoding.

2.3 Iterative Decoding

The maximum likelihood decoding has the smallest error probability. However, it is not practical because it requires an exponential amount of time with respect to the code length. An important property of LDPC codes is that they have simple suboptimal decoding algorithms. These algorithms are iterative and run in $O(n)$ time, where n is the block length. They are called message-passing algorithms. In these algorithms, messages are exchanged between variable nodes and check nodes iteratively. In each iteration, every check node c receives messages from all its neighbor variable nodes (two vertices are neighbors if they are adjacent). Based on these messages, the check node computes new messages and sends them to its neighbors. A message that the check node c sends to a variable node v is a function of the incoming messages from all neighbors of c except v. Similarly, variable nodes send messages to their neighbor check nodes. After enough iterations, the decoder decides on the value of a variable node based on the messages received at the node.

The most common message-passing algorithm is the *belief propagation*. We now briefly describe this algorithm. We use terminology similar to that in [54]. Suppose the codeword $\underline{X} = (x_1, x_2, ..., x_n) \in \{-1, +1\}^n$ is transmitted through a binary channel and the vector $\underline{Y} = (y_1, y_2, ..., y_n)$ is received. Let m_0 be the Log likelihood ratio (LLR) of a received bit conditioned on the observation of the channel output for that bit. In other words, we define

$$m_0 = \ln \frac{p(x_i = +1|y_i)}{p(x_i = -1|y_i)} = \ln \frac{f_Y(y_i|x_i = +1)}{f_Y(y_i|x_i = -1)}. \tag{4}$$

Let $m_{vc}^{(l)}$ denote the message sent from a variable node to its incident check node c. Also let $m_{cv}^{(l)}$ denote the message that the check node c sends to its incident variable node v. As defined in [54], we define

$$\gamma : [-\infty, +\infty] \to GF(2) \times [0, \infty], \tag{5}$$

where $GF(2)$ is the binary field and

$$\gamma(x) = (\gamma_1(x), \gamma_2(x)) = (\text{sgn}(x), -\ln \tanh|\frac{x}{2}|), \tag{6}$$

$$sgn(x) = \begin{cases} 0 \text{ if } x < 0 \\ 0 \text{ with proability} \frac{1}{2} \text{ if } x = 0 \\ 1 \text{ with probability} \frac{1}{2} \text{ if } x = 0 \\ 1 \text{ if } x > 0 \end{cases} \tag{7}$$

Then, the update equations under belief propagation are:

$$m_{vc}^{(l)}(x) = \begin{cases} m_0 \text{ if } l = 0 \\ m_0 + \sum_{c' \in C_v \setminus c} m_{c'v}^{(l)} \text{ if } l > 0 \end{cases} \text{ and} \tag{8}$$

$$m_{cv}^{(l)} = \gamma^{-1}(\sum_{v' \in V_c \setminus v} \gamma(m_{v'c}^{(l-1)})) \tag{9}$$

where C_V is the set of check nodes adjacent to the variable node v and V_C is the set of variable nodes adjacent to the check node c. Here, $C_v \setminus c$ denotes the exclusion of the member c from the set C_V. If l_m is the total number of iterations, the decoder evaluates $LLR_v = m_0 + \sum_{c' \in C_v} m_{c'v}^{(l_m)}$ for a variable node v. The variable node is decoded to 1 if $LLR_v > 0$ and to -1 otherwise.

2.4 Properties of the Iterative Decoding

The iterative decoder has three important properties: concentration, convergence to the cycle-free case and density evolution and threshold effect. These properties hold asymptotically and were first proved for the binary erasure channel [71] and later generalized to a very broad class of channels and decoding algorithms [112]. Here, we state the results given in [112].

Concentration: We define $P_e^n(l)$ as the expected fraction of incorrect messages that are passed in the lth iteration where the expectation is over all instances of the code, the choice of the message, and the realization of the noise [112]. For any positive δ, the probability

that the actual fraction of incorrect messages in the lth iteration for any particular such instance lies outside the interval $(P_e^n(l) - \delta, P_e^n(l) + \delta)$ converges to zero exponentially fast with respect to n, the code length.

Convergence to the cycle-free case: Let $P_e^\infty(l)$ be the expected fraction of incorrect messages passed in the lth iteration assuming that the graph does not contain cycles of length $2l$ or less. Then, $P_e^n(l)$ converges to $P_e^\infty(l)$ as n goes to infinity.

Density evolution and threshold determination: We can compute $P_e^\infty(l)$ by a method called density evolution. In addition, there exists a channel parameter $\sigma^\star$, the threshold, with the following property. If $\sigma < \sigma^\star$ then $\lim_{l\to\infty} P_e^\infty(l) = 0$. On the other hand, if $\sigma > \sigma^\star$ then there exists a constant $\xi(\sigma) > 0$ such that $P_e^\infty(l) > \xi(\sigma)$ for all $l \geq 1$.

2.5 Density Evolution

Richardson et al. developed an algorithm, called *density evolution* to find the densities of the messages exchanged between variable nodes and check nodes [112], [54]. In this method, the distributions of messages from variable nodes to check nodes at two consecutive iterations of belief propagation are connected by a recursive formula. They used this method to determine the performance of LDPC codes and to find optimum degree distributions. We now briefly describe density evolution.

Density evolution is applicable to memoryless binary-input output-symmetric (MBIOS) channels. An MBIOS channel is a discrete-time channel whose input X is $+1$ or -1 and whose output Y depends only on the current input symbol and satisfies

$$f_Y(y|X = 1) = f_Y(-y|X = -1). \tag{10}$$

For these channels the performance of the LDPC code is independent of the encoded data. Thus, to find the performance of the code ensemble, we may assume the all-zero codeword is being sent through the channel (we assume that zero is mapped to 1 and one is mapped to -1 prior to the transmission, thus the transmitted codeword is $(x_1, x_2, ..., x_n) = (1, 1, ..., 1)$). Let $f : \mathbb{R} \longmapsto \mathbb{R}$ be an integrable function. We define

$$\lambda(f) = \sum \lambda_i f^{\otimes(i-1)} \tag{11}$$

where $\otimes$ denotes convolution. Let P_l and Q_l be the densities of the random variables $m_{vc}^{(l)}$ and $m_{cv}^{(l)}$, respectively, provided that the all-zero codeword has been sent. Then, the formulas for density evolution can be written as

$$P_l = P_0 \otimes \lambda(Q_l) \tag{12}$$

$$Q_l = \Gamma^{-1}(\rho(\Gamma(P_{l-1}))) \tag{13}$$

where Γ is defined in [54] in the following way. If z is a random variable with distribution F_z, then the distribution of $\gamma(z)$ is defined as

$$\Gamma(F_z)(s,x) = I_{(s=0)}\Gamma_0(F_z)(x) + I_{(s=1)}\Gamma_1(F_z)(x) \tag{14}$$

where I is the indicator function and

$$\Gamma_0(F_z)(x) = 1 - F_z^-(-\ln\tanh(\frac{x}{2})), \Gamma_1(F_z)(x) = F_z(\ln\tanh(\frac{x}{2})). \tag{15}$$

Now, we can use these formulas to find the densities of the messages. It is shown in [54] that the error probability goes to zero if and only if P_l converges to a dirac delta at infinity as $l \to \infty$. Thus, using the above formulas we can determine whether for a given channel parameter σ the LDPC code ensemble can have an arbitrarily small error rate. The threshold $\sigma^\star$ is the supremum value of σ such that the error rate can be made arbitrarily small.

Now we are left with an optimization problem. We need to optimize λ and ρ to make the threshold $\sigma^\star$ as large as possible. It is mentioned in [54] that if we let only a few nonzero coefficients in ρ and λ polynomials, we can find very good degree distributions.

Chung developed a more efficient method called discretized density evolution to implement the density evolution [24]. Chung et al. also introduced a Gaussian approximation method [25], which is much simpler than the original density evolution. It is shown that for the long block lengths (higher than 10^4), the optimized codes have performance close to the Shannon limit [54], [70], [24]. However, the design of good short-length LDPC codes is still a challenging problem.

2.6 Other Developments

Here we briefly state some other important results in the area of random LDPC codes. Some results on bounding the performance of LDPC codes were provided by [78], [18], [114], and [8]. Distance distributions and stopping set distributions along with some other asymptotic properties were found in [66], [67], [19], and [83]. A major development appeared by the results on design and analysis of finite-length LDPC codes over the BEC [27], [109], and [111].

Finally, we note that there are several deterministic constructions of LDPC codes. One of the most important constructions is based on finite geometries [63]. The codes constructed deterministically are usually easier to implement; however, they normally have inferior performance to that of the random codes. In this research we are only concerned with random constructions.

2.7 Wireless Sensor Networks

Recently, the rapid technology development in materials science, MEMS, and bioengineering systems has made sensing technologies readily available. The convergence of sensor technologies, communications, and computing has emerged to provide the potential to overcome the barriers of time, scale, and environment. Distributed sensor networks are becoming a feasible solution to various data collection applications such as military sensing and tracking, environment monitoring, patient monitoring and tracking, and learning environments.

Wireless sensor networks have received interest lately, with proposed applications in military and civilian surveillance and sensing tasks and potential services that would enhance the ability of the growing domain of wireless technologies [4]. Wireless sensor networks have benefited from advances in both MEMS technology and networking.

The problem of wireless sensor networks considers a large number (in the order of thousands) of identical nodes which possess limitations in available energy, computational power, memory, and communication range. In potential sensing applications, the sensor nodes may be randomly deployed in a hazardous or dangerous environment where the nodes are physically inaccessible after deployment. In this way, the design of the network needs to consider

energy conserving schemes to account for a limited energy supply, low memory/computation and resilient networking schemes to account for the hazardous environment.

The primary task of wireless sensor networks is to have the sensors relay information back to one or more base stations. This is accomplished without globally known network addressing (i.e. IP addresses). Therefore, the sensor nodes rely on broadcasting techniques to deliver information in possibly a multihop fashion (because of limited communication range, there is usually no direct transmission from sensors to the base). Information is either sent from the base stations to the sensor nodes or from the sensor nodes to the base stations. The flow of information in wireless sensor networks distinguishes itself from ad hoc networking and other varieties of wireless networking. The characteristics of the other wireless communications is less defined, as communications may occur in node-to-node communications which potentially requires communication schemes and algorithms that are quite different than those of wireless sensor networks.

The design of sensor networks [3] must consider routing protocols and communication schemes to best fit the intended sensing task at hand. Furthermore, these schemes must observe the restrictions of the sensor network such as the conservation of energy while still maintaining a certain level of resiliency against node failure or capture.

Resiliency in large-scale sensor networks is linked to the connectivity of the network. That is, is every node in the network able to communicate with the base stations in the network. Without such connectivity, the network is unable to provide proper functionality. Moreover, redundancy that is added through sending information through multiple paths is also another function within sensor networks that is utilized.

We are interested in design and analysis of sensor networks using graph and information theoretic tools. Sensors are usually densely scattered over a field. Thus, the number of sensors is usually large. Related problems have been studied in the context of random graph theory [10], continuum percolation and geometric probability [77, 86], and the study of wireless network graphs [11, 12, 32, 42, 44, 65, 115, 124, 125]. In random graph theory, the model $G(n, p)$ is extensively studied, in which edges appear in a graph of n vertices with probability p independent of each other. In continuum percolation theory, usually infinite

graphs on $\mathbb{R}^d$ are studied. Finally, in geometric probability and the study of graphs of wireless networks, large-scale graphs over the plane are usually studied.

In [44], the connectivity of large-scale wireless networks is studied. In [65], [124], and [93], k-connectivity of wireless networks has been studied. In [65], k-connectivity is studied in the context of fault-tolerant networks. In [124] authors study the asymptotic critical transmission radius for k-connectivity and asymptotic critical neighbor number for k-connectivity in wireless networks. In [93], we studied connectivity and k-connectivity for large-scale sensor networks. In that paper, we specifically studied the effects of node and link failures and the distribution function of the nodes on connectivity properties of sensor networks. The connectivity in ad-hoc and hybrid networks is studied in [31]. In [30], trade-off between connectivity and capacity of dense networks is studied. In particular, the effect of the attenuation function on network properties is studied. Medium access (MAC) layer capacity of wireless ad hoc networks has been studied in [7]. The transport and information theoretic capacity has been studied extensively, for example see [41–43, 64, 68, 92]. However, almost all previous analytic results consider graphs in which the number of nodes tend to infinity.

There are also many papers on the empirical study of network characteristics. For example, a survey on routing protocols for wireless sensor networks can be found in [2]. Although many of these papers, consider practical-size networks, they usually rely on simulations. Simulations are a crucial and useful tool for the study of wireless networks; however, as it is discussed in the paper, they are not enough. Thus, it is very important to have an analytical framework for design and study of wireless networks.

CHAPTER III

DECODING PROBLEMS FOR LDPC CODES

3.1 Introduction

In this chapter we study decoding of LDPC codes [101], [100], [94], [98], [95]. We first consider the case where LDPC codes are used over the binary erasure channel (BEC). Then, we generalize the results for other symmetric channels. The application of LDPC codes over BEC has been studied extensively [27, 70], [119], [118], [84]. When the message passing algorithm is applied to an LDPC code over the BEC, it results in a very fast decoding algorithm [70]. However, the performance of this decoder is inferior to that of the maximum likelihood (ML) decoder. Here, we first derive some bounds on the performance of the ML decoder over the BEC. We then propose a technique to improve the performance of the message passing decoder while keeping the speed of the decoding fast.

Asymptotic analysis of the performance of LDPC codes has been done successfully [112], [70], [54]. Capacity achieving degree distributions for the binary erasure channel have been introduced in [70], [118], [119] and [84]. Although using the asymptotic analysis we can find good degree distributions, generating good finite-length LDPC codes has always been a challenge. In fact, in many practical applications we have to use short-length or moderate-length (finite-length) codes. Finite-length analysis of LDPC codes over the BEC was accomplished in [27]. In that paper, authors also proposed to use the finite-length analysis in order to find good finite-length codes for the BEC. Here, we take a different approach. Instead of trying to find good LDPC codes we improve the decoding of the existing codes. The combination of the optimized codes using the finite-length analysis and the improved decoding algorithm that we present in this chapter can result in good coding schemes over the BEC. Although the method we propose can be applied to any code length, its impact is more important for finite-length codes because for large values of code-lengths there exist codes that achieve the capacity of the BEC. Thus, here we concentrate

more on the moderate-length and short-length codes. More specifically, we consider the lengths that are less than or equal to 10^4. We then generalize the improved algorithm for other memoryless binary-input output-symmetric (MBIOS) channels [94]. We finally study stopping sets in LDPC codes. It is shown that stopping sets play a crucial role in decoding performance of LDPC codes. We show that finding stopping sets in LDPC codes is an NP-hard problem.

Throughout the chapter we assume the following terminology. By a graph we mean a simple graph, i.e., a graph with no loops (edges joining a vertex to itself) and no multiple edges (several edges joining the same two vertices). However, a multigraph may have loops or multiple edges. Let A be a subset of the vertices in the graph g. $N(A)$ shows the set of neighbors of A in g. More generally, for $j \in \mathbb{N}$, $N^j(A)$ is the set of vertices in g from which there is path of length j to a vertex in A. Let D be a subgraph of g such that its vertex set is A. We say D is induced by A if D contains all edges of g that join two vertices in A. Let $e = vw$ be an edge in the graph g. By contracting e we mean that we identify the vertices v and w and remove all the resulting loops. Note that unlike the usual definition of contraction, we do not remove the duplicate edges resulting from identifying v and w. Let D be a subgraph of g. If we contract all the edges in D, we say that we have contracted D into a vertex. Let G be a bipartite multigraph with bipartition $V(G)$ and $C(G)$, For any $A \subseteq V \cup C$ we define $I_g(A)$ as the graph induced by the vertices in A and their neighbors. For a set E, 2^E is the set of all subsets of E. For a graph g, $deg_g(v)$ is the degree of v in g. If V is the set of vertices in g and $U \subseteq V$, then $deg_U(v)$ is the number of neighbors of v in U. For a random variable X, we show its distribution by $F_X(x)$. If the random variable has a well-defined density function, we represent the density function by $f_X(x)$. Similar to [54], we define $P_e(F_X) = \Pr\{X < 0\} + \frac{1}{2}\Pr\{X = 0\}$.

3.2 Bounds on the Performance of ML Decoding over the BEC

The ML decoding has the best possible bit error rate. Since we are concerned with improving the iterative decoding of LDPC codes, the ML decoding gives us the best possible

improvement we may get. Thus it is useful to study the ML decoder and its properties. Some properties of the ML decoding of LDPC codes have been studied before, see for examples [72], [78], [18] and [27]. We first consider the asymptotic capacity of LDPC codes over the BEC under maximum likelihood decoding. LDPC codes can be defined by their Tanner graphs [121]. Consider the ensemble $g(\lambda, \rho)$ of bipartite graphs defined by their degree distributions. We define the ML capacity (threshold) $\epsilon^\star$ of the ensemble as the supremum value of the parameter ϵ such that a randomly chosen code from the ensemble can achieve an arbitrarily small bit error rate for sufficiently large n almost surely over a BEC with an erasure probability ϵ. There is no concentration result known for the ML decoding of LDPC codes over general MBIOS channels. However, the ML capacity is well defined and is always greater than or equal to the threshold of the iterative decoding. With the above definition we can find simple lower and upper bounds on the ML capacity of an ensemble. Since these upper and lower bounds are very close to each other (they are practically the same at least for regular codes), they provide an estimate of the ML capacity of the codes. A simple lower bound can be found using the union bound given in [27] and the asymptotic distance distributions of the regular LDPC codes derived in [66]. Note that in [19] authors have independently derived the lower bound in the context of error exponent of ML decoding. Let $g(d_v, d_c)$ be the ensemble of the regular LDPC codes with variable nodes and check nodes of degrees d_v and d_c, respectively. Then, we have the following lower bound [101].

Theorem 1. *Let $t(d_c, \theta)$ be the only positive root of*

$$\frac{(1+t)^{d_c-1} + (1-t)^{d_c-1}}{(1+t)^{d_c} + (1-t)^{d_c}} = 1 - \theta \tag{16}$$

Define $p(d_v, d_c, \theta)$ as

$$p(d_v, d_c, \theta) = \begin{cases} \frac{d_v}{d_c} \ln\left(\frac{(1+t)^{d_c}+(1-t)^{d_c}}{2t^{\theta d_c}}\right) - d_v H(\theta) & \text{if } d_c \text{ is even or } \theta \in (0, \frac{dc-1}{d_c}) \\ -\infty & \text{otherwise} \end{cases} \tag{17}$$

Then, the ML capacity of the ensemble $g(d_v, d_c), d_v > 2$ is lower bounded by the supremum value of ϵ such that $\epsilon H(\frac{\theta}{\epsilon}) + p(d_v, d_c, \theta) < 0$ for all $\theta \in [0, \epsilon]$.

Proof. Let h_i be the i'th column of an $m \times n$ parity-check matrix H. Suppose a codeword is transmitted over a BEC channel with the erasure probability ϵ and let ε denote the set of variable nodes that correspond to the erased bits. Further assume that $\epsilon < \epsilon_l^\star$, where $\epsilon_l^\star$ is the lower bound for the ML capacity given by the theorem. Let B_η be the event that $|\frac{|\varepsilon|}{n} - \epsilon| < \eta$. We have $\text{pr}(B_\eta) = 1 - o(1)$ for all $\eta > 0$. The ML decoder can decode the received word correctly if the columns of H that correspond to the erased bits are independent. Let I be the set of indices of the erased bits. We define

$$A_l = \{(h_{i_1}, h_{i_2}, ..., h_{i_l}) : h_{i_1} \oplus h_{i_2} ... \oplus h_{i_l} = 0, i_1 < i_2 < ... < i_l, i_j \in I, j = 1, 2, ..., l\} \quad (18)$$

for $l = 1, 2, ..., n$. Let X_l be a random variable defined by $X_l = |A_l|$, where $|.|$ shows the cardinality of a set. The typical minimum distance of LDPC codes from $g(d_v, d_c)$ with $d_v > 2$ increases linearly with the code length. More specifically, for the ensemble $g(d_v, d_c)$ with $d_v > 2$ there exists $\delta(d_v, d_c) > 0$ such that the probability that the minimum distance of a randomly chosen code from the ensemble is less than or equal to $n\delta(d_v, d_c)$ converges to zero as $n \to \infty$ [38], [28]. Thus we conclude that

$$\text{pr}\Big[\exists l \in \{1, 2, ..., n\delta(d_v, d_c)\}, X_l > 0\Big] = o(1) \quad (19)$$

Let p_l be the probability that l randomly chosen columns of H sum to zero. Therefore

$$E(X_l||\varepsilon| = e) = \binom{e}{l} p_l \quad (20)$$

As it is shown in [66] $\lim_{n\to\infty} \frac{1}{n} \ln p_{\theta n} = p(d_v, d_c, \theta)$. Therefore, $\lim_{n\to\infty} \frac{1}{n} \ln E(X_{\theta n}|B_\eta)$ approaches $\epsilon H(\frac{\theta}{\epsilon}) + p(d_v, d_c, \theta)$ as η goes to zero. Let $c_s = \sup_{\delta(d_v,d_c) \le \theta \le \epsilon + \eta} (\epsilon H(\frac{\theta}{\epsilon}) + p(d_v, d_c, \theta))$. Using $\epsilon < \epsilon_l^\star$, it is easy to show that for sufficiently small η we have $c_s < 0$. Thus

$$E(X_{\theta n}|B_\eta) = O(e^{-n|\frac{c_s}{2}|}). \quad (21)$$

Now define $X = \sum_{l=n\delta(d_v,d_c)}^{n(\epsilon+\eta)} X_l$. By the above discussion we have :

$$E(X|B\eta) = \sum_{l=n\delta(d_v,d_c)}^{n(\epsilon+\eta)} EX_l = O(n\epsilon e^{-n|\frac{c_s}{2}|}) \to 0 \text{ as } n \to \infty. \quad (22)$$

Since $X \in \{0, 1, ...\}$, using $E(X|B_\eta) \to 0$ and the Markov inequality we conclude that $\Pr(X = 0|B_\eta) = 1 - o(1)$. Since $\text{pr}(X = 0) = \text{pr}(X = 0|B_\eta).\ \text{pr}(B_\eta) + \text{pr}(X = 0|B_\eta^c).\text{pr}(B_\eta^c)$

we obtain $\text{pr}(X = 0) = 1-\text{o}(1)$. Combining this with (19) we conclude that the ML decoder can decode the received word successfully almost always. □

Let $\epsilon_l^\star$ be the lower bound for the ML capacity given by Theorem 1. As an example, $\epsilon_l^\star$ of the $g(3, 6)$ ensemble is equal to $\epsilon_l^\star = .483$ while the capacity under message passing (the threshold found using density evolution) is $\epsilon_0 = .429$. Since, we always use LDPC codes below their threshold we conclude that for sufficiently large code lengths, the ML decoder is likely to decode the received word even though the message passing decoder fails.

Using the distance distributions of the irregular codes, it is possible to generalize the above argument for the irregular codes. The distance distributions of irregular codes have been found by several authors [1], [67] and [19]. As it is shown in [28] if $\lambda_2\rho'(1) < 1$ the minimum distance of the expurgated ensemble increases linearly with the code length. Thus, we find the following bound for the expurgated ensemble.

Theorem 2. *Consider the ensemble $g(\lambda, \rho)$ that satisfies $\lambda_2\rho'(1) < 1$. Let $b_\theta = b_\theta(\lambda, \rho)$ be the average distance distribution as defined in [66] . Then the ML capacity of the expurgated ensemble $g(\lambda, \rho)$ is lower bounded by the supremum value of ϵ such that $\epsilon H(\frac{\theta}{\epsilon}) - H(\theta) + b_\theta(\lambda, \rho) < 0$ for all $\theta \in [0, \epsilon]$.*

It is useful to find an upper bound for the ML capacity. The bound would obviously be an upper bound for the iterative decoder as well. First, using the Markov's inequality the following lemma can be easily concluded.

Lemma 1. *Let $\theta > 0$ be a constant. Let also N be a positive integer such that for all $n > N$, the sequence of random variables Z_n, where $0 \le Z_n \le n$ satisfies $E(Z_n) \le (\theta+o(1))n$. Then, for all $\alpha > 0$ there exist N' and $\delta > 0$ such that for all $n > N'$ we have $pr\{Z_n < (\theta+\alpha)n\} > \delta$.*

Let ξ_i be the fraction of variable nodes of degree i and φ_i be the fraction of check nodes of degree i. Let us define $\Psi(x) = 1 - \sum_i \xi_i(1-x)^{i-1}(1+(i-1)x)$ and $\Phi(x) = \sum_i \varphi_i x^i$. By a simple observation, we can find the following upper bound for the capacity of LDPC codes over the BEC [101].

Theorem 3. *To have arbitrarily small bit error probability under the ML decoding on a BEC with erasure probability ϵ, we must have*

$$1 - R \geq \frac{\epsilon[1 + \Psi((1-\epsilon)^{d_{c_{max}}-1})]}{1 - \Phi(1-\epsilon)} \tag{23}$$

Proof. Let us construct a matrix H' from the parity matrix H by selecting each column of H with probability $(1-\epsilon)$ independently and replacing it with the zero vector. The nonzero columns of H' correspond to the erased bits. If we pick a row from H at random, this row is equal to zero in H' with probability $\Phi(1-\epsilon)$. Therefore, on the average we have $m\Phi(1-\epsilon)$ zero rows in H'. We now prove the following lemma to complete the proof.

Lemma 2. *If we peak a random integer j between 1 and n, then with probability at least $\epsilon\Psi((1-\epsilon)^{d_{c_{max}}-1}) - o(1)$ we have at least two rows in H' such that their j'th element is one and all their other elements are zero.*

Proof. Call the required probability p_j. For clarity of exposition consider the regular ensemble $g(d_v, d_c)$. Let F_j be the event that the j'th column in H' be nonzero, thus $\text{pr}\{F_j\} = \epsilon$. There are d_v rows $i_1, i_2, ..., i_{d_v}$ in H whose j'th element is one. Let I_j be the set consisting of these rows. Thus we have $|I_j| = d_v$. Let EV_{i_k} be the event that only the j_k'th element of the i'th row in H' is equal to one. Since $d_v < \infty$, with high probability the positions of the ones in all of the rows in I_j do not overlap except for the j'th position. Therefore, given that the j'th column is preserved in H', the events EV_{i_k} are independent. Since the probability of any event is less than or equal to one, to obtain p_j, it suffices to consider only the case when EV_{i_k}'s are independent and add an $o(1)$ to the result . Now, obviously we have $\text{pr}\{EV_{i_k}|F_j\} = (1-\epsilon)^{(d_c-1)}$. Considering the above discussion it is easy to show that

$$p_j = \epsilon\Big\{1 - \{[1-(1-\epsilon)^{d_c-1}]^{d_v} + d_v[1-(1-\epsilon)^{d_c-1}]^{d_v-1}(1-\epsilon)^{d_c-1}\} - o(1)\Big\} \tag{24}$$

For irregular codes using the following inequality

$$\prod_{i=1}^{n}(1-x_i) + \sum_{i=1}^{n} x_i \prod_{j\neq i}(1-x_i) \leq (1-x_1)^n + nx_1(1-x_1)^{n-1} \tag{25}$$

For $0 \leq x_1 \leq x_2 \leq ...x_n \leq 1$, we obtain

$$p_j \geq \epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) - o(1) \tag{26}$$

□

Thus we showed that if we pick a random integer i between 1 and n, with probability at least $\epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) - o(1)$ we have at least two rows in H' such whose i'th element is one and all the other elements are zero. Any such i reduces the rank of H' by at least one. Therefore, if we define $Z_n := \text{rank}(H')$, we have

$$\begin{aligned} E(Z_n) = E(\text{rank}(H')) \leq m - m\Phi(1-\epsilon) - n\epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) + o(1)n = \\ n\{(1-R)[1-\Phi(1-\epsilon)] - \epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) + o(1)\}. \end{aligned} \tag{27}$$

By defining

$$\theta := (1-R)[1-\Phi(1-\epsilon)] - \epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) \tag{28}$$

we have

$$E(Z_n) \leq (\theta + o(1))n. \tag{29}$$

Now we show that for an arbitrarily small error probability we must have $\epsilon \leq \theta$. Suppose $\epsilon > \theta$. As we showed $E(Z_n) = E(\text{rank}(H')) \leq (\theta + o(1))n$. Let $0 < \kappa < \epsilon - \theta$ be a constant. Then, by Lemma 1, there exist N' and $\delta > 0$ such that for all $n > N'$ we have $\text{pr}\{Z_n < (\theta+\kappa)n\} > \delta$. Therefore, with a strictly positive probability that is independent of n we have $\text{rank}(H') < (\theta+\kappa)n$. Hence the decoder can find the value of at most $(\theta+\kappa)n$ erasures. Consequently, at least $(\epsilon - \theta - \kappa)n$ erasures remain after the decoding. This implies that the overall error probability of the decoder is at least $(\epsilon - \theta - \kappa)\delta$ .Therefore, for reliable communication we must have $\epsilon \leq \theta$. Thus

$$(1-R)[1-\Phi(1-\epsilon)] - \epsilon\Psi((1-\epsilon)^{d_{cmax}-1}) \geq \epsilon. \tag{30}$$

This completes the proof of the theorem.

□

In the above argument if we just consider the rows that are zero in H' and omit the discussion about the rows with weight one, we will get a slightly weaker bound as

$$1 - R \geq \frac{\epsilon}{1 - \Phi(1-\epsilon)}. \tag{31}$$

This is the same bound given in [118] for the iterative decoding of the LDPC codes. Note that in [118], the bound is derived for the iterative decoding but the above discussion shows that the bound is also valid for the ML decoding. It is also noteworthy that examining the proof of Theorem 3, we find out that the upper bound given by (31) is valid for any individual code in the ensemble while the one given in (23) is valid for typical codes. This is because we made use of the cycle-free neighborhood assumption. As an example, for the ensemble $g(3,6)$, Theorem 3 gives the upper bound $\epsilon = .489$ while (31) provides a slightly weaker bound $\epsilon = .491$. Furthermore, the authors in [114] have also obtained the upper bound $\epsilon = .491$, that is again slightly weaker than the one given by Theorem 3. However, their bound has been proven for the individual codes.

Figures 3 and 4 show the upper bound and the lower bound that are given by Theorems 1 and 3 for $g(3,d_c)$ and $g(4,d_c)$ respectively. As the figures suggest the two bounds are practically the same. As an example for irregular graphs, we consider the ensemble of LDPC codes defined by

$$\lambda(x) = .142696x + .562771x^2 + .294532x^{10}, \qquad \rho(x) = x^6. \tag{32}$$

Using Theorems 2 and 3 we find that the ML capacity for the given ensemble satisfies $.4899 \leq \epsilon \leq .4948$. It is worth noting that for the calculation of the lower bound we used [1] to find the weight spectrum of the code. Again we conclude the bounds are sufficiently tight. Therefore, we can approximate the ML capacity from the given bounds. Note that Theorem 3 gives an upper bound that is easily computable for any degree distributions. However, for the lower bound we need to have the distance distribution of the code.

Since we are interested in the finite-length LDPC codes, it is desirable to choose the codes that satisfy $\lambda_2\rho'(1) < 1$. This is because as is shown in [28] and [109] if we have $\lambda_2\rho'(1) > 1$, then the minimum stopping set and the minimum distance will be sublinear

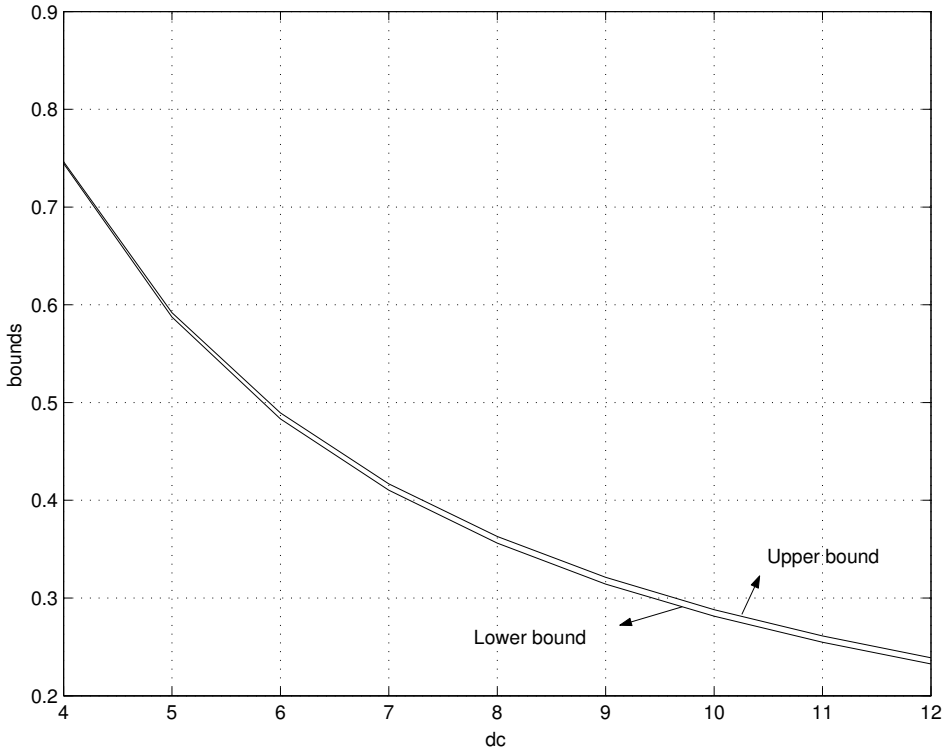

Figure 3: Lower and upper bounds for the ML capacity of $g(3, d_c)$.

with high probability and therefore we will have small stopping sets in the graph which is not desirable for finite length codes. In the above definition we defined the ML threshold for the bit erasure probability. It is mentioned in [27] that the threshold for bit erasure probability and block erasure probability may be different. Any upper bound for the threshold for bit erasure probability is an upper bound for the threshold for block erasure probability as well. Therefore, the upper bound in Theorem 3 is also an upper bound for the threshold for the block erasure probability. In the ensembles for which $\lambda_2\rho'(1) < 1$, the thresholds for the block erasure probability and the bit erasure probability are the same. This is because for these ensembles there is no codeword with weight less than or equal to $n\delta(d_v, d_c)$ with a high probability. Therefore, if the ML decoder cannot decode the received word, there will be at least $n\delta(d_v, d_c)$ erasures that are left after the decoding is performed. Thus, the bit error rate is at least $\delta(d_v, d_c)$ times the block error rate. The same argument works for the iterative decoder if we replace the minimum distance by the size of the minimum

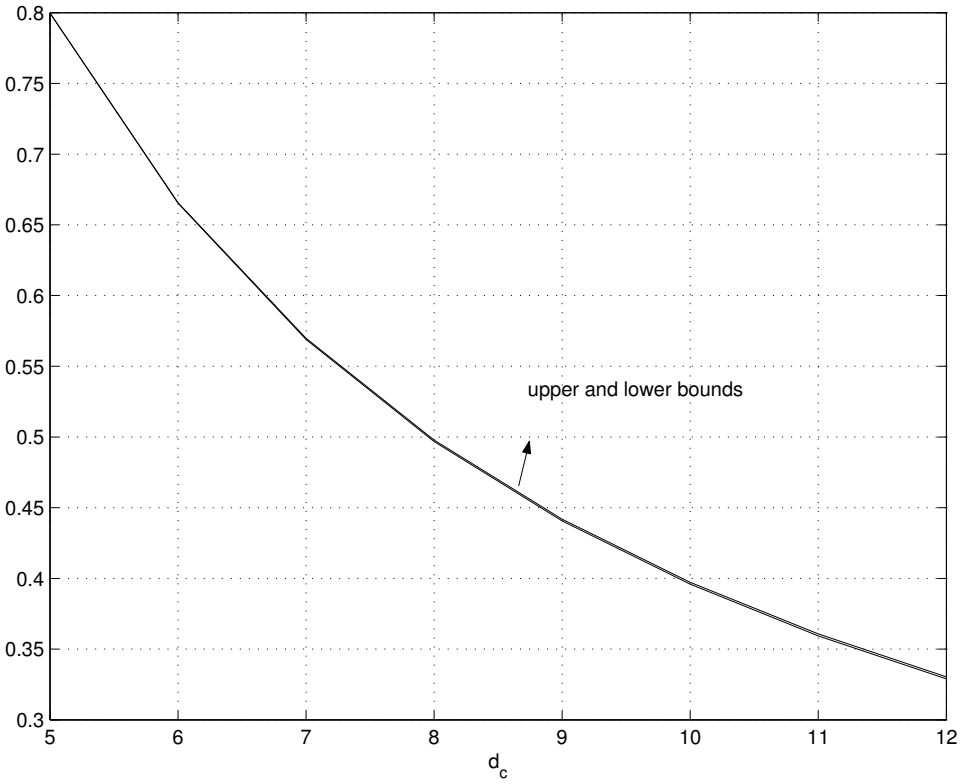

Figure 4: Lower and upper bounds for the ML capacity of $g(4, d_c)$.

stopping set. Consequently, the lower bounds in Theorems 1 and 2 are also valid for the threshold for block erasure probability. In fact, although we stated the theorem for the bit error probability threshold, in the proof we showed that the block error rate goes to zero.

3.3 Improving the Iterative Decoding

3.3.1 Description of Algorithms

The iterative decoding of LDPC codes over the BEC is much faster than the ML decoding. However, it has higher error probability. Our aim in this section is to decrease the error probability while keeping the decoding fast. We mostly focus on moderate and short-length codes. We use the message passing algorithm with some modifications. Let $V = V(g) = \{v_1, v_2, ..., v_n\}$ and $C = C(g) = \{c_1, c_2, ..., c_m\}$ be the set of variable and check nodes, respectively. A stopping set S is defined in [27] as a subset of V such that all neighbors of S are connected to S at least twice. Let ε be the subset of the set of variable nodes that is erased by the channel. It is proved in [27] that the iterative decoding fails if and only if

$\mathcal{E}$ contains a stopping set. In [27] it is also shown that the set of remaining erasures when the decoder stops is equal to the unique maximal stopping set of $\mathcal{E}$.

Let $T_A(n)$ be the average time required for the standard iterative decoding of an LDPC code of length n when it is used over a BEC with the erasure probability ϵ. Let B be an improved decoding method for the same code when used over the same channel. Let $T_B(y,n)$ be the time that algorithm B needs to decode a received word y and let $T_B(n)$ be the average time of the decoding of the code using algorithm B. We want to have:

$$T_B(n) \leq (1+\gamma)T_A(n) \tag{33}$$

$$\forall y, \quad T_B(y,n) \leq CT_A(n) \tag{34}$$

where γ is a small constant close to zero and C is a sufficiently small constant. Our simulations show that the algorithm we propose in this section (algorithm C) will achieve the above inequalities with $\gamma < .05$ and $C < 10$.

Theoretically, any LDPC code has a threshold ϵ_{th} such that if $\epsilon > \epsilon_{th}$ then the error probability of the standard iterative decoding is bounded away from zero by a strictly positive constant. On the other hand if $\epsilon < \epsilon_{th}$, an arbitrarily small error probability is attainable if n, the length of the code , is large enough [112], [70]. However, for finite-length codes the situation is deferent. First, we may get an error floor and cannot decrease the error probability as we want. Moreover, to decrease the error probability ,for example from 10^{-3} to 10^{-6}, we need to decrease ϵ by a considerable amount. Here we propose a method for decoding LDPC codes over BEC that has the same complexity as the message passing decoder. However, its error rate is considerably smaller.

The key idea is the following observation [101]. Consider a BEC with an erasure probability ϵ and an LDPC code of length n that has a small enough error probability. If the message passing decoder fails to decode a received word completely, then there exists a very small number (usually less than or equal to 3 bits) of undecoded bits that if their values are exposed to the decoder, then the decoder can finish the decoding successfully. Note that this is true only when the bit error rate is small enough (for example less than 10^{-2}). Simulations and intuitive arguments strongly confirm the above statement for different LDPC

codes.

Let us recall the message passing decoding of LDPC codes over the BEC [70]. The algorithm can be stated as:

- For all unlabelled check nodes do the following. If the values of all but one of the variable nodes connected to the check node are known, set the missing variable bit to the XOR of the other variable nodes and label that check node 'finished'. If all the variable nodes connected to the check node are known label the check node as finished.The procedure is done sequentially, i.e, one check node at a time.

- Continue the above procedure until all check nodes are labelled as finished or the decoding cannot continue further.

We want to improve the above algorithm. Let us call the above algorithm A and the first improved algorithm B. For the erasure patterns that algorithm A finishes the decoding successfully, both algorithms are the same. The difference between the two algorithms is when algorithm A fails to complete the decoding of a received codeword. In this case algorithm B continues the decoding as following. It chooses one of the unknown variable nodes w_1 (we will discuss how to choose this variable node) and guesses its value (for example by setting its value to zero). Now it continues as follows.

- For all unlabelled check nodes do the following: If the value of all but one of the variable nodes connected to the check node are known, set the missing variable bit to the XOR of the other variable nodes and label it as a finished check node. If all the variable nodes connected to the check node are known then if the check node is satisfied label that check node 'finished', otherwise label it 'contradicted'. The procedure is done sequentially, i.e, one check node at a time.

- Continue the above procedure until all check nodes are labelled or the decoding cannot continue further.

Once the above procedure is finished, if all of the check nodes are labelled and none of them is labelled contradicted, the decoder outputs the resulting word as the decoded word.

If all of the check nodes are labelled but some of them are labelled contradicted, then it changes the value of w_1, the guessed variable node, and repeats the decoding from there. This time the decoding finishes successfully because we have found the actual value of w_1. But if the decoding stops again (i.e. some of the check nodes are not labelled) we have to choose another unknown variable node, w_2, and guess its value to continue the decoding. Again, if some check nodes are labelled as contradicted, we have to go back and try other values for w_1 and w_2. Obviously, algorithm B is efficient only if the number of guesses is very small. Fortunately, simulation results show that even if we limit the number of guesses to a very small number, we can decrease the error rate by a considerable amount. Thus, in practice we limit the number of guesses to a maximum value g_{max}. If after g_{max} guesses the decoding does not finish, we claim a decoding failure. In fact, simulations show that with the right choices of the variable nodes to guess, usually the decoding finishes successfully by one or two guessed variable nodes. Note that the above algorithm does not need any extra computation other than the usual iterative decoding. Thus the decoding is very fast.

Now let us consider the problem of choosing the variable nodes w_i's that we need to guess their values. One easy method is to choose them from the set of variable nodes with the highest degree. Note that a variable node of degree d is present in d equations. Therefore when we assume that its value is known, any parity-check equation that has only one unknown variable node other than the guessed variable node will free a variable node. Therefore, intuitively we expect that guessing a high-degree variable node results in freeing more unknown variable nodes. However, we can still improve our method of choosing w_i's as following. First we choose a high-degree unknown variable node w_j and examine its neighborhood. If guessing w_j frees at least f unknown variable nodes for a suitable constant f, we accept w_j as one of our guesses. Otherwise we choose another high-degree variable node. In our simulations we chose the value of f between $.5d_{c_{max}}$ and d_{cmax}.

Algorithm B has two problems. First, the complexity of the algorithm grows exponentially with the number of guesses. Although the number of guesses is very small, this is undesirable. In fact, if the complexity of the algorithm increased linearly with the number of guesses we could increase g_{max} and decrease the error probability substantially. Second, it is

possible that the algorithm declares a wrong word as the output of the decoding. However, this can happen only if the ML decoder cannot decode the corresponding codeword. Since the ML decoder has a very low error probability, this happens with a very small probability. We now propose algorithm C that copes with both problems [101].

Let w_1 be the first variable node that we guess. Let x_1 be the value of w_1. From now on any variable node whose value is determined by the algorithm can be represented in one of the following forms: x_1, $\overline{x_1} = x_1 \oplus 1$, 1 or 0 . In general, if the algorithm makes g guesses, any variable node that is determined after the first guess can be represented as

$$a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus ... \oplus a_g x_g \tag{35}$$

where $a_j \in \{0, +1\}$. Therefore, any variable node that is determined after the first guess can be represented by $(a_0, a_1, ...a_g)$. After the first guess, algorithm C continues as follows:

- For all unlabelled check nodes do the following: If the values of all but one of the variable nodes that are connected to the check node are known, compute the value of $(a_0, a_1, ...a_g)$ for the missing bit and label that check node as finished. If all the variable nodes that are connected to the check node are known then label that check node as a basic equation. The procedure is done sequentially, i.e, one check node at a time.

- Continue the above procedure until all check nodes are labelled or the decoding cannot continue further.

If it is necessary, algorithm C makes other guesses. If after the maximum possible number of guesses some of the variable nodes are unlabelled, then we claim decoding failure. Now suppose after $g \leq g_{max}$ guesses all the check nodes are labelled. Now we have the following.

Lemma 3. *The received word is ML decodable if and only if the set of basic equations have a unique solution.*

Proof. By the labelling procedure, any choice for the values of $x_1, x_2, ..., x_g$ satisfy all the parity check equations that are labelled finished. Therefore, decoding is possible if and only if a unique choice of $x_1, x_2, ..., x_g$ satisfies all the basic equations. □

Therefore, after all the check nodes are labelled, we examine the set of basic equations. If they have a unique solution, we determine $x_1, x_2, ..., x_g$ and then we find the values of all the variable nodes. Otherwise, we claim decoding failure. Note that since g is a very small number, solving the basic equations is a very simple task and can be done quickly. In fact, it is easy to show that algorithm C has complexity $O(g_{max}^2 n)$. Sometimes, although the set of basic equations do not have a unique solution, they still determine a subset of $\{x_1, x_2, ..., x_g\}$ uniquely. In this case we can replace the values of these variable nodes in the expressions for unknown variable nodes and consequently we may be able to recover some of the bits. This approach is specifically useful when we deal with a code that has error floor due to the small minimum distance. Note that the procedure of finding a variable node for guessing in algorithm C is the same as algorithm B. The following lemma determines the number of basic equations:

Lemma 4. *Let $\mathcal{E}$ be the subset of the set of variable nodes that are erased by the channel and S be the unique maximal stopping set in $\mathcal{E}$. Then the number of basic equations is equal to:*

$$N_B(S) = |N(S)| - |S| + g \tag{36}$$

Proof. At the beginning of the guessing process there are $|N(S)|$ unlabelled check nodes. Any of these check nodes is either a basic check node or determines exactly one variable node. Since there are $|S| - g$ variable nodes that are determined by the check nodes, $|N(S)| - |S| + g$ check nodes are labelled as basic equations. □

Note that algorithm C is equivalent to the ML decoder if we do not limit the maximum number of guesses, g_{max}. In fact, in this case algorithm C is just an efficient implementation of the ML decoder. It is worth noting that the set of basic equations depends on the choice of variable nodes we guess.

An alternative method is to perform ML decoding on the remaining erasures whenever the iterative decoding fails. This decoding is much faster than the ordinary ML decoding and has exactly the same performance as the ML decoding. However, it has two problems. First, it does not satisfy the requirement in (47), because generally the ML decoding of LDPC codes over the BEC has time complexity $\Theta(n^3)$. Second, the ordinary ML decoding of LDPC codes requires $\Theta(n^2)$ space while algorithms A, B and C require $O(n)$ space.

3.3.2 Bounds on the Number of Guesses in Algorithms B and C

Here we study the required number of guesses by algorithm B and C based on the properties of the Tanner graph of the code. Instead of providing asymptotic results, we focus on graph theoretic results that we think are useful in finite-length analysis. To do that we need to extend the iterative decoding to bipartite multigraphs. Let G be a bipartite multigraph with bipartition $V(G)$ and $C(G)$, where $V(G) = \{v_1, v_2, ..., v_n\}$ and $C(G) = \{c_1, c_2, ..., c_m\}$ are the sets of variable and check nodes, respectively. The iterative decoding works as follows on G. At the beginning, all the erased variable nodes are labelled unknown and the following algorithm is repeated in each step:

- If only one of the edges that is connected to a check node is incident with an unknown variable node, label that variable node as known.

Note that the above algorithm is not a decoding algorithm for a real code. We just define it to make our discussion simpler. Let give some more definitions. We define a set $B \subset V(G)$ to be sufficient if by knowing the values of the variable nodes in B, the iterative decoder can finish the decoding successfully. A set $B \subset V(G)$ is called unnecessary if $V(G) - B$ is sufficient. In other words, $B \subset V(G)$ is unnecessary if the iterative decoder can determine the values of erased variable nodes, when the variable nodes in B are erased but all the other variable nodes are known . Obviously, a set B is unnecessary if it does not contain a stopping set.

Assume the iterative algorithm A fails to decode a received word on a graph g. Let S be the stoping set that remains after the decoding stops and let $F = I_g(S)$. We define an equivalence relation R on $V(F)$ in the following way. We write vRw if there is a path from

v to w on F that does not contain any check node of a degree higher than two. Obviously, R is an equivalence relation. Thus this relation partitions $V(F)$ into $p(F)$ equivalence classes. Let $A_1, A_2, ..., A_{p(F)}$ be the equivalence classes of R. Note that if the value of v is exposed to the decoder, then the decoder can find the values of all variable nodes in the equivalence class $[v]$ of v.

For a bipartite graph F we construct the graph $R(F)$ as follows. For each equivalence class A_i we contract all the variable nodes in A_i and all the check nodes that do not have any neighbors outside A_i into one vertex u_i. Figure 5 shows an example of this construction. Note that every check node in $R(F)$ has a degree at least three. Assume the iterative algorithm A fails to decode a received word Y on a graph g. Let S be the stoping set that remains after the decoding fails and let $F = I_g(S)$. We have the following.

Theorem 4. *Let Y be ML decodable and Z be the random variable that is equal to the number of guesses that is required by algorithm B to finish the decoding. Let also $M(F)$ be the minimum sufficient set in $R(F)$ and $p(F)$ be the number of equivalent classes in $v(F)$. Then, we have*

$$|M(F)| \leq Z \leq p(F), \tag{37}$$

and the lower bound is always attainable by using the right choices of the guessed variable nodes.

Proof. After each guess in algorithm B, all the variable nodes in at least one of the equivalence classes will be determined. Therefore, after $p(F)$ guesses the values of all the variable nodes are found. Let $M(F) = \{u_j : j \in I_F\}$ be the minimum cardinality sufficient set in $R(F)$. Suppose, in each step we choose a variable node in one of A_j's such that $j \in I_F$. Then after $|M(F)|$ steps all the variable nodes are determined. Obviously the number of guesses cannot be less than $|M(F)|$.

□

For example in Figure 5, $\{A_1\}$ is the minimum cardinality sufficient set in $R(F)$. Therefore, we have $|M(F)| = 1$. If if we choose our first guess from the vertices in A_1, only one

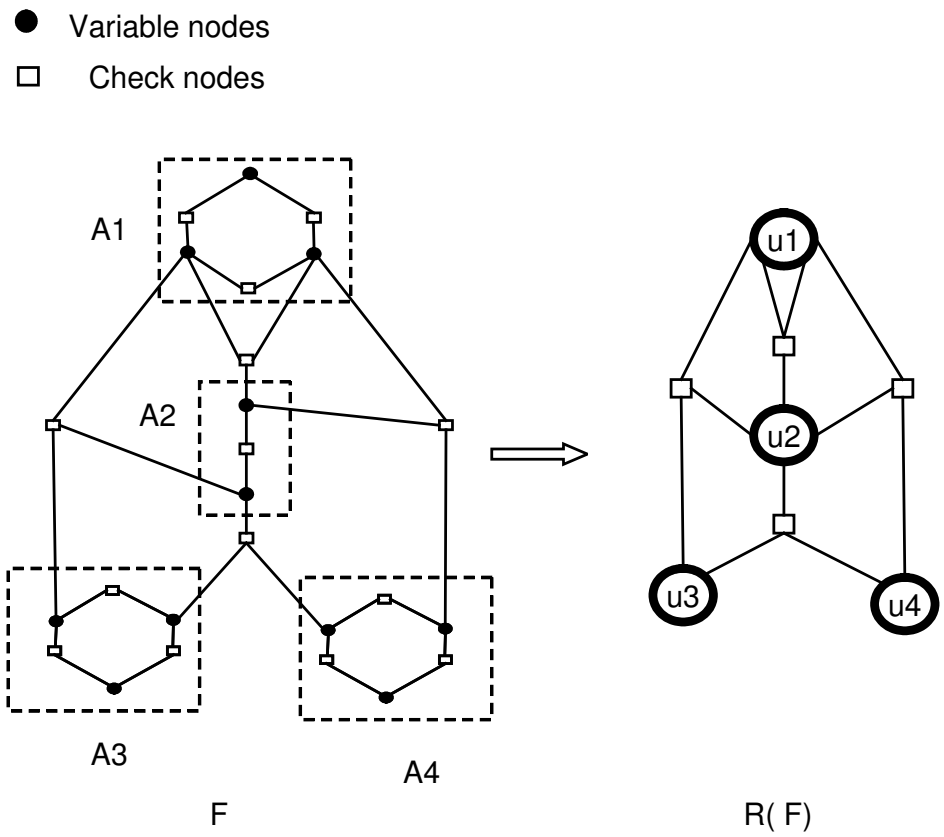

Figure 5: Construction of $R(F)$.

guess is enough to finish the decoding successfully. Since $p(F) = 4$ the number of guesses satisfies $1 \leq Z \leq 4$. However, it is easy to show that the number of required guesses is always less than or equal to 2 for this example .

The above discussion shows the connection between the graph theoretic properties and the number of guesses. It would be nice if we can use these arguments to obtain some probabilistic results such as finding the average number of guesses for a specific code and channel. Note that we deal with a finite-length analysis and asymptotic analyses are not useful. In fact, we need some discussion based on the BER of the iterative decoder for a finite-length code to find the probabilistic properties of the given algorithm.

3.3.3 Improving Algorithms B and C by Reduction of Number of Guesses

Let g be the Tanner graph of an LDPC code and Z be the number of guesses required by algorithm B or C when the iterative algorithm A fails. Later we will show that Z is

very small (usually $E(Z) < 2$). However, we can still reduce the number of guesses. Here, we introduce one method to reduce Z. We first need to give some results based on the two-edge-connected components of the Tanner graph of LDPC codes.

Assume that all the vertices in g, the Tanner graph of the code, have a degree at least two. For any $A \subseteq V \cup C$ we defined $I_g(A)$ as the graph induced by the vertices in A and their neighbors. The graph $I_g(A)$ is sparser than the graph g in the sense that the degree of each vertex in $I_g(A)$ is less than or equal to the degree of that vertex in g. Let $S \subseteq V$ be a stopping set. Then, obviously $I_g(A)$ has at least one cycle because any vertex in $I_g(A)$ has a degree at least two. Let h^j be the j'th row of H. Suppose $x = (x_1, x_2, ..., x_n)$ is a codeword and x_j be the bit corresponding to the variable node v_j. Any row h^i can be written as a parity check equation in the form

$$E^i = \sum_{j=1}^{n} h_{ij} x_j = 0. \tag{38}$$

where h_{ij} is the element in i'th row and j'th column of H. Consider the case that the iterative decoder fails but the ML decoder can decode the received word. Let S_m be the set of variable nodes that the iterative decoder cannot decode. Consider a variable node $v_t \in S_m$. In this case there exist $I \subseteq \{1, 2, ..., m\}$ and $U \subset V \backslash S_m$ such that

$$\sum_{j \in I} E^j = x_t + \sum_{x_j \in U} x_j. \tag{39}$$

We say that the set of parity checks corresponding to $\{h^j : j \in I\}$ frees the variable node v_t and we call this set of parity checks a freeing set for v_t. Let A be a subgraph of g. We define $C(A)$ as the set of parity check nodes in A. For $k \geq 2$, we say a graph is k-edge-connected if it has at least two vertices and no set of at most $k-1$ edges separates it. We have the following theorem [101].

Theorem 5. *Assume $S \subset V$ be a nonempty stopping set such that the ML decoder can decode the word when the set S is erased. Suppose we receive a word for which S is the unique maximum stopping set. Then there exists a two-edge-connected subgraph of $I_g(S)$, say g_S, such that the set of parity checks in g_S ,i.e. $C(g_s)$, frees an erased variable node.*

Proof. Let y be the word constructed from x by marking the variables in S as erasures. Since the ML decoder can decode y, there exists a freeing set for any of the variable nodes in S. Let v_t be an arbitrary variable node in S and F be a minimum freeing set for v_t. Define $D = I_{I_g(S)}(F)$, then D is a connected graph. Otherwise, D has at least two components, A and B. Note that all the variable nodes in D except one (the variable node v_t) have even degrees in D. Thus in at least one of the two subgraphs A and B (assume A), all the variable nodes have even degrees. Therefore, $F - C_g(A)$ is a freeing set for v_t. This contradicts the assumption that F is minimum (note that $C_g(A) \neq \varnothing$).

If D is two-edge-connected we are done so we may assume D is not two-edge connected. We consider two cases as shown in figure 6.

- Case1: $\deg_D(v_t) > 1$, where $\deg_D(v)$ denotes the degree of the vertex v in the graph D. In this case, the degree of any variable node in D is at least two. By our assumption and definition of D the degree of any check node in D is at least two. Since the degree of each vertex in D is at least two, D contains at least one cycle. Thus, D contains at least one two-edge-connected component. Let D' be a graph obtained by contracting any two-edge-connected component of D to a vertex. Then, D' is a tree. This is because if D' had a cycle then that cycle would be in a two-edge-connected component and would have been contracted to a vertex. Note that D' has at least two vertices otherwise D would be two-edge-connected. Since D' is a tree, it has at least two leaves (vertices of degree one). Moreover, since $\deg_D(v) > 1$ for all $v \in V(D)$, these leaves must correspond to two-edge-connected components C_1 and C_2 in D. Since C_1 and C_2 are disjoint, for at least one of them, say C_1, we have $v_t \notin C_1$. Next we show that $C(C_1)$ is a freeing set for a variable node. Additionally C_1 is two-edge-connected. The existence of C_1 proves the theorem.

  Let $e = uw$ be the only edge that is connected to C_1 in D'. Assume $u \in C$ and $w \notin C$. If w is a variable node then $C(C_1)$ will free w because w has degree one in $I_g(C_1)$ and all the other variable nodes in $I_g(C_1)$ have even degree. This is because the degrees of these variable nodes in D and C_1 are the same. Now if w is a check node, then u

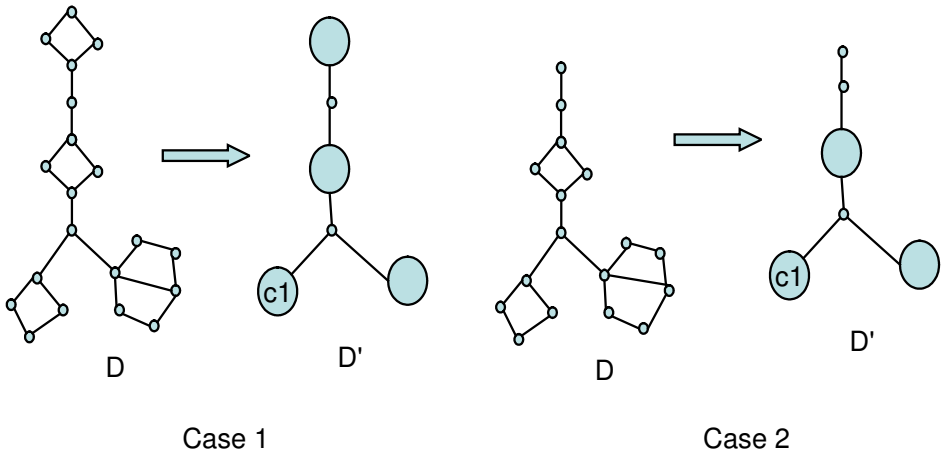

Figure 6: Construction of the graph D'.

is a variable node. Then $C(C_1)$ will free u. This is because by the above assumption $u \neq v_t$. Therefore $\deg_D(u)$ is even and $\deg_{C_1}(u) = \deg_D(u) - 1$ is an odd number.

- Case2: $\deg_D(v_t) = 1$. We construct the graph D' Similar to case 1 and v_t will be one of the leaves in D' . But D' has at least one more leaf that corresponds to a two-edge-connected component. Call this leaf C_1. Obviously $v_t \neq C_1$. Therefore, using the argument in Case 1, $C(C_1)$ is a freeing set for a variable node.

figure 6 shows the above argument. □

The immediate result of Theorem 5 is the following corollary.

Corollary 1. *If we append all the parity-check equations that are formed by adding the parity-check equations in the two-edge-connected subgraphs of g to H, then the iterative decoding on the new H is equivalent to the ML decoding.*

Obviously, this is not feasible because there are lots of such equations. However, we will show in next sections that we can exploit Theorem 5 in order to improve the iterative decoding.

Corollary 2. *If we apply the message passing algorithm to an acyclic graph it will be equivalent to the ML decoding.*

This is a well known result that can be proved for the erasure channel as following. Let S be the stopping set that remains after the iterative decoding. For any set c^* of check nodes in $I_g(S)$, the graph that is induced by the vertices in c^* and their neighbors in $I_g(S)$ has at least two leaves. Since any check node in c^* has a degree at least two in $I_g(S)$, these leaves must be variable nodes. Therefore, the set c^* cannot be a freeing set for any variable node. Note that in any acyclic graph we have at least two variable nodes of degree one (we are assuming that check nodes have always degrees greater than one) and any stopping set in the graph must contain at least two variable nodes of degree one.

Let $C(g) = \{c_1, c_2, ..., c_m\}$ be the set of check nodes in g, the Tanner graph of the parity-check matrix H. We define the set $T \subseteq 2^{C(g)}$ as following. For any set $R = \{c_j : j \subseteq \{1, 2, ..., m\}\}$, we have $R \in T$ if and only if $I_g(R)$ is a two-edge connected subgraph of g. Let $EQ(H)$ be the set of parity-check equations that are obtained by adding the set of parity-check equations from an element of T. Recall from Theorem 5 that the equations in $EQ(H)$ are sufficient for ML decoding. However, the number of these equations is extremely high and we cannot use all of them in the iterative decoding. Note that all of these equations are redundant because they are obtained by adding some parity-check equations in H . However, it turns out that by using a very small number of suitably chosen equations from this large set of equations, we can reduce the number of guesses in algorithms B and C. Again we use these equations whenever the algorithm A fails. Note that any two-edge connected graph is composed of several cycles. A cycle is the simplest two-edge connected graph. Here, we only consider short cycles. Let C_{2l} be the number of cycles of length $2l$ in $g(d_v, d_c)$. It is shown in Appendix that the expected value of C_{2l} is equal to

$$E(C_{2l}) = \binom{n}{l}\binom{m}{l}\frac{l!(l-1)!}{2}\left\{\frac{[d_v d_c(d_v-1)(d_c-1)]^l}{E \times (E-1)...(E-2l+1)}\right\} \tag{40}$$

where $m = (1-R)n$ is the number of check nodes and $E = nd_v = md_c$ is the number of edges in the graph. For a constant number l we have

$$\lim_{n\to\infty} E(C_{2l}) = \frac{\left[(d_v-1)(d_c-1)\right]^l}{2l} \tag{41}$$

and for $0 < \theta < \alpha = 1 - R$

$$c(\theta) = \lim_{n\to\infty} \frac{1}{n} \ln(E(C_{2\theta n})) = H(\theta) + \alpha H(\frac{\theta}{\alpha}) + \theta \ln[d_v d_c (d_v - 1)(d_c - 1)] - d_v H(\frac{2\theta}{d_v}) - 2\theta \ln 2 \tag{42}$$

where $H(x) = -x\ln(x) - (1-x)\ln(1-x)$. Therefore, the average number of the finite-length cycles is a constant and does not increase with n. Obviously, the same argument works for irregular codes. In order to reduce the number of guesses in algorithm B or C we find some parity-check equations that construct short cycles (cycles of lengths four or six) in the Tanner graph and add them together to find new parity-check equations. Note that these equations are used only if the iterative decoding algorithm A fails. We will show that using a small number of these equations suffices to reduce the number of guesses. Using (41) we can find the expected number of equations that we need to consider. For example in $g(3, 6)$, if we just take parity-check equations that construct cycles of lengths four or six, on the average we will find around 192 equations.

3.3.4 Simulation Results

In this section we provide some empirical results. First, we experimentally verify the main claim that we made in Section 3.3.1 (i.e. a very few number of guesses is enough to finish the decoding). We then give simulation results for the LDPC codes of lengths $n = 1000$ and $n = 10,000$ and evaluate the performances of the algorithms A, B and C. We compare these algorithms based on the bit error rate, average speed and the speed of decoding for a specific received word. Notice that the LDPC codes that we use here are not optimized for the corresponding length and rate. We did the simulations for half rate codes and for each length we picked a code with reasonable performance and degree distributions. In fact, we observed that for a fixed length and a maximum degree, the relative performances of the algorithms are roughly independent of the degree distribution of the code.

Let us first study the number of guesses in algorithms B and C when the algorithm A fails. Again, let us define the random variable Z to be the number of required guesses when the standard iterative decoding (algorithm A) fails to decode a received word. For the length $n = 1000$ we considered the following degree distribution:

$$\lambda_1(x) = .0769x + .6923x^2 + .2308x^5, \qquad \rho_1(x) = .4615x^5 + .5385x^6 \tag{43}$$

To evaluate the number of required guesses we set $g_{max} = \infty$ and decoded 10^{10} bits that were transmitted over the BEC with the erasure probability $\epsilon = .36$. Figure 7 shows the empirical probability density function for the number of required guesses. We note that in more than 70 percent of the cases for which the iterative decoder fails, only one guess is enough to complete the decoding successfully. We also note that the number of required guesses is always less than or equal to 8. The error rate of algorithm A is about 10^{-5}. Even if we limit the maximum number of guesses to 4, using algorithms B or C we can improve the error rate by almost two orders of magnitude.

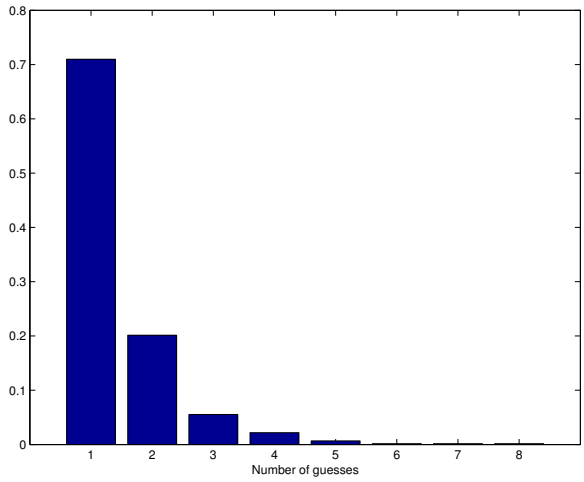

Figure 7: Distribution of the number of guesses that is required for successful decoding at $\epsilon = .36$.

However, the situation changes if we increase ϵ. For example, at $\epsilon = .39$, the average bit error rate of the standard iterative decoder (algorithm A) is .0039. Figure 8 shows the empirical probability density function for the number of required guesses. The figure shows that the number of required guesses increases as we increase ϵ. For example, if we limit the maximum number of guesses g_{max} to 4, we can decrease the bit error rate by only one order of magnitude using algorithms B or C. However, it is worth noting that even for $\epsilon = .39$, the average number of required guesses is still very small, ($g_{av} = 2.23$).

Let us now examine the effect of the code length on the number of guesses. We picked

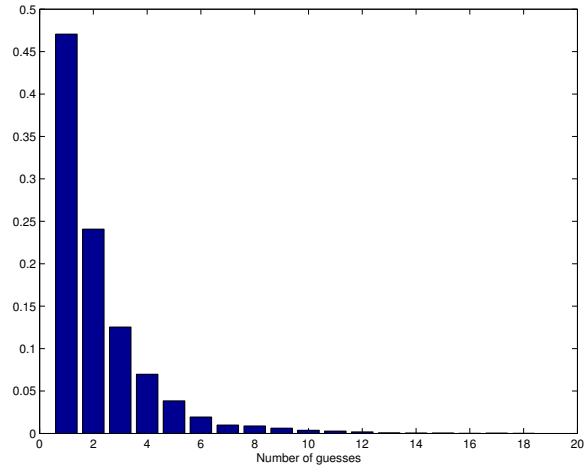

Figure 8: Distribution of the number of guesses that is required for successful decoding at $\epsilon = .39$.

an LDPC code of length 10^4 with the following distributions:

$$\lambda_2(x) = .4706x^2 + .2353x^7 + .2941x^{29}, \qquad \rho_2(x) = .7843x^9 + .2157x^{10} \tag{44}$$

Figure 9 depicts the empirical probability density function of Z for this code. Note again that the number of guesses is largely concentrated on the small values (less than or equal to 4). Therefore, we conclude that the number of required guesses is small for most of the practical code lengths. Therefore, algorithms B and C are efficient. Note that in the above example, when the standard iterative decoding fails, there are about 3500 erasures left when the decoding stops. However, in most of the cases by knowing the values of less than or equal to 4 erased bits the decoder can find the value of all the 3500 erasures!

Now we examine the performance of the proposed algorithms. Note that algorithms B and C have almost the same bit error rate. In all simulations we set g_{max} to 6. Figure 10 shows the performance of algorithms A and C for a code from the ensemble $g(\lambda_1, \rho_1)$ with the length 1000. The figure shows that the gap between the bit error rate of the two algorithms increases as ϵ decreases. At $\epsilon = .4$, the bit error rate of algorithm A is twenty times bigger than the bit error rate of algorithm C. This gap increases to about three orders of magnitude when ϵ is reduced to .36. This suggests that algorithm C can alleviate the error floor problem in LDPC codes. Specifically, this algorithm can be very

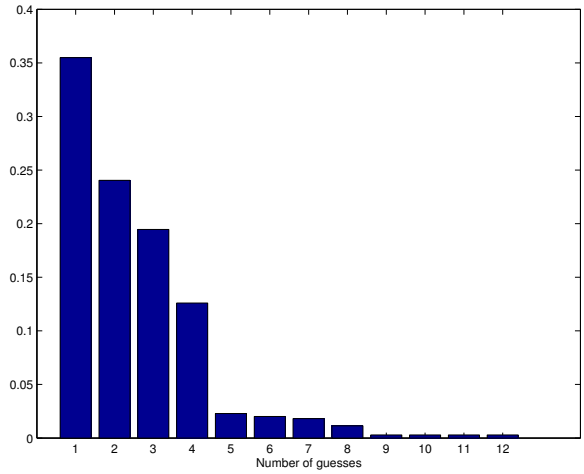

Figure 9: Distribution of the number of guesses that is required for successful decoding at $\epsilon = .39$ and code length $= 10^4$.

useful when very small bit error rate is required. On the other hand, for the large values of ϵ the improvement due to algorithm C becomes negligible. Our simulations show that for the values of ϵ that the bit error rate of the iterative decoder is less than or equal to 10^{-2}, a good improvement is possible in the bit error rate by applying algorithm C.

There are several stopping sets in a Tanner graph of an LDPC code. Some stopping sets are weak in the sense that if the values of one or two bits in the stoping set is exposed to the iterative decoder, the decoder can finish the decoding successfully. For these stopping sets $|M(F)| = |M(I_g(s))|$ is a very small number. On the other hand, some stopping sets are very strong and their $|M(F)|$ is a large number. When ϵ is close to 1,we usually face with strong stopping sets. For example at $\epsilon = 1$ the strongest stopping set (i.e. $V(g)$) occurs. As we decrease ϵ, the strong stopping sets become less probable while the weak sopping sets become more probable. Therefore, fewer number of guesses is required to finish the decoding. This discussion explains why the gap between the bit error rates of algorithms A and C increases as ϵ decreases.

Table 1 shows the average number of required guesses by algorithm C for the received blocks that algorithm A fails. The table suggests that the average number of guesses is very small. Note that the values in Table 1 are slightly smaller than the average values obtained by Figures 7 and 8 because for those diagrams we have $g_{max} = \infty$, but Table 1 is obtained

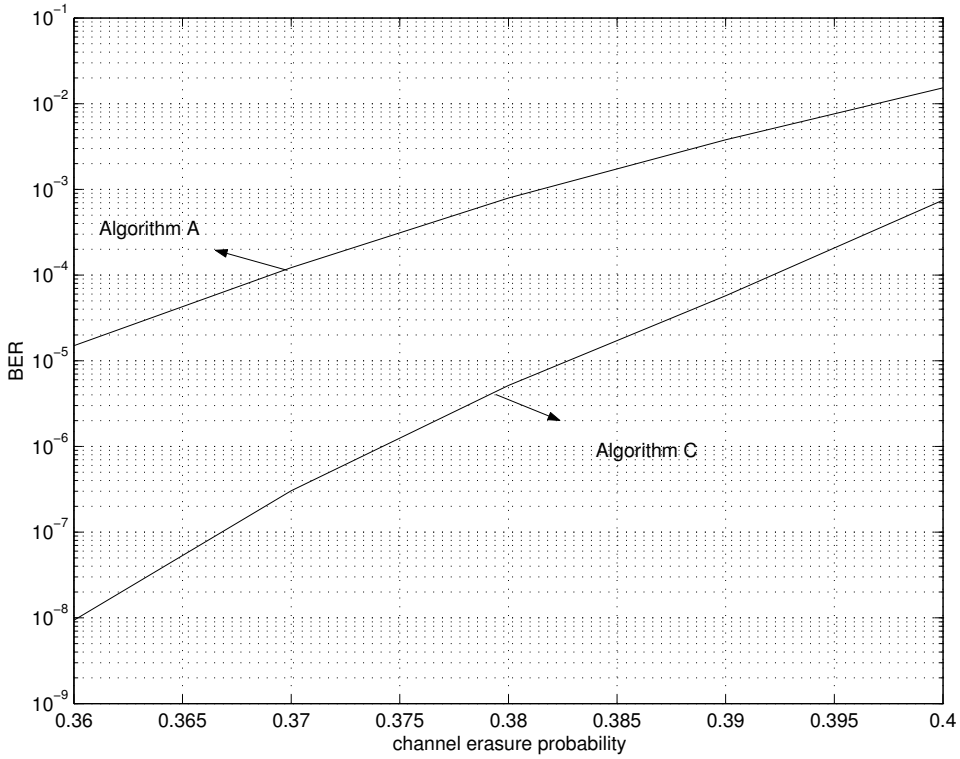

Figure 10: comparisons of the bit error rates of algorithms A and C for code length $n = 10^3$.

for $g_{max} = 6$.

Table 1: The average number of guesses for the LDPC code of length 1000.

ϵ	Average number of guesses
.36	1.38
.37	1.59
.38	1.78
.39	2.11

Figure 11 shows the performance of algorithms A and C for a code from the ensemble $g(\lambda_2, \rho_2)$ with a length of 10^4. We see that the results are similar to that of the code with the length 1000. The above results are obtained when $g_{max} = 6$.

We note that, in both ensembles $g(\lambda_1, \rho_1)$ and $g(\lambda_2, \rho_2)$, the minimum distance grows linearly with the code lengths. In fact, when we generated codes from these ensemble, we made sure that these codes did not have small stopping sets that could cause error floor. In order, to see the effect of error floor on algorithms B and C, we generated a code of length

10^4 from the ensemble defined by

$$\lambda_3(x) = .2223x + .3884x^2 + .1934x^7 + .1959x^{14}, \quad \rho_3(x) = .88x^6 + .22x^7 \tag{45}$$

We note that for this ensemble we have $\lambda_3\rho'(3) = 1.5154 > 1$. Thus, with high probability, a code generated from this ensemble has a small minimum distance and shows error floor. Figure 12 shows the performance of algorithms A and C for the code. We observe that for high bit error rates, algorithm C shows some improvement over algorithm A. However, as we approach the error floor region, the improvement decreases. This is because, for this code even the ML decoder shows an error floor (because of small minimum distance) and thus using improved decoding methods, does not help much in the error floor region. Table 2 shows the average number of guesses for this code.

Table 2: The average number of guesses for the LDPC code of length 10000 that has an error floor.

ϵ	Average number of guesses
.450	2.34
.455	2.78
.460	3.12
.465	3.67

Now we present some experimental results for the running time of the algorithms. Clearly, these results are dependent on the specific computer program and the platform we use. However, a relative timing comparison can be made from these simulations. We give the results for a code of length 1000 from the ensemble $g(\lambda_1, \rho_1)$ and a code of length 10^4 from the ensemble $g(\lambda_2, \rho_2)$. The erasure probability of the channel ϵ is chosen such that the bit error rate of algorithm A is 10^{-3}. In all cases we decoded 10^{10} bits and measured the average running time and the maximum running time for the decoding of received blocks. Let $T_A(n), T_B(n)$, and $T_C(n)$ show the average time of decoding of the LDPC code of length n from the given ensembles using algorithms A, B and C, respectively. Table 3 shows the relative average time of algorithms B and C with respect to the standard iterative decoding. From the table we conclude that the average running time of all the above algorithms are almost the same.

Recall that we defined $T_B(y, n)$ as the time that algorithm B needs to decode a received

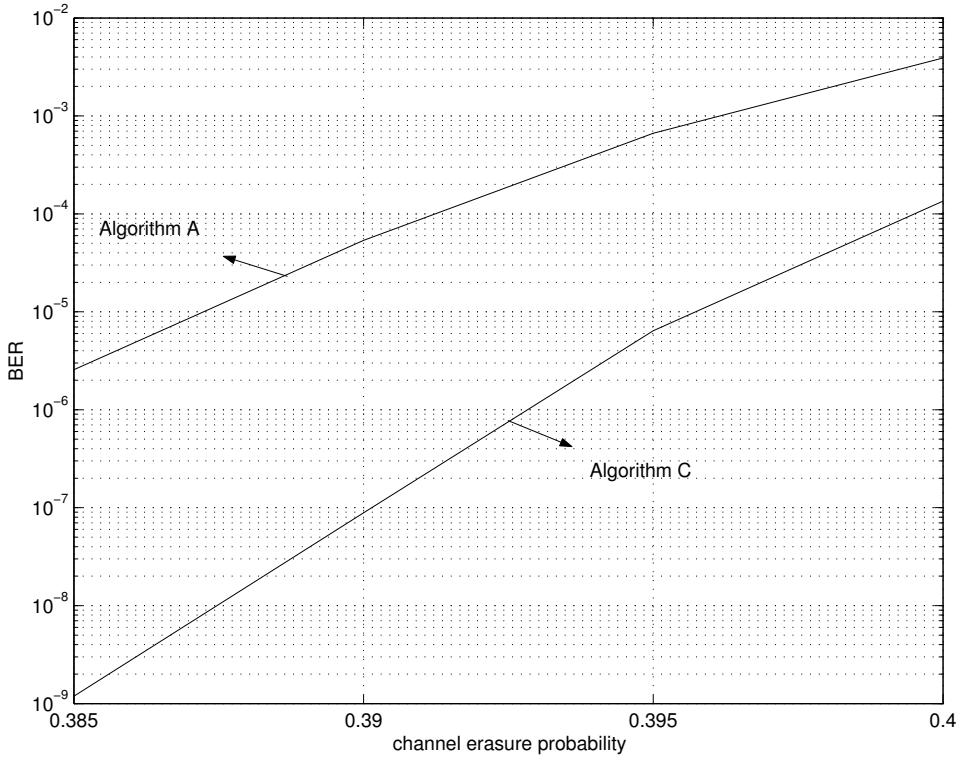

Figure 11: Comparisons of the bit error rates of algorithms A and C for a code of length $n = 10^4$.

word y. Let $T_B^m(n)$ show the maximum value of $T_B(y, n)$ over all the received blocks. We define $R_B(n) = \frac{T_B^m(n)}{T_A(n)}$. $R_C(n)$ is defined similarly. Table 4 shows the values of $R_B(n)$ and $R_C(n)$. The table suggests that although algorithm B has a good average running time, it can be very slow for some specific received blocks. However, the running time of algorithm C for any received block is always less than 10 times the running time the standard iterative decoder. Combining this with the fact that the average running time of algorithm C is almost the same as the iterative decoder we conclude that algorithm C is efficient in terms of running time. Since algorithm C is fast and has a bit error rate smaller than the standard iterative decoder, it can be considered as an efficient way of decoding LDPC codes over the BEC.

Table 3: Comparison of average running time of different algorithms

	$n = 1000$	$n = 10^4$
$\frac{T_B(n)}{T_A(n)}$	1.06	1.08
$\frac{T_C(n)}{T_A(n)}$	1.01	1.02

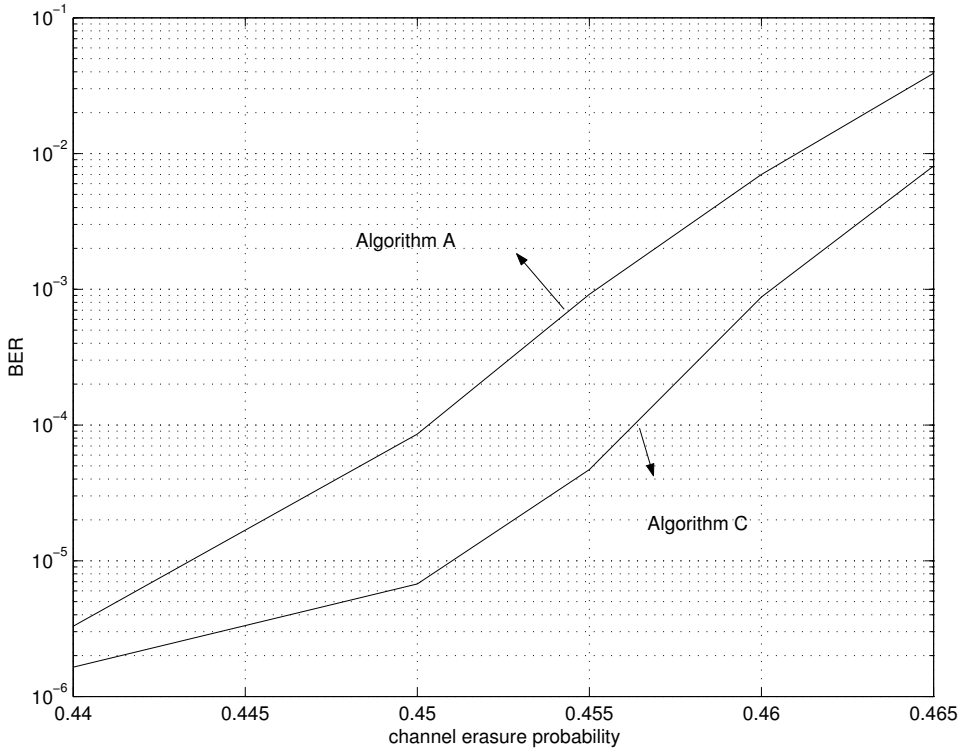

Figure 12: Comparisons of the bit error rates of algorithms A and C for a code of length $n = 10^4$ that has an error floor.

Table 4: Maximum ratio of the running times of algorithms B and C to the running time of algorithm A.

	$n = 1000$	$n = 10^4$
$R_B(n)$	67.2	71.3
$R_C(n)$	6.2	9.7

As we mentioned, it is possible to reduce the average number of guesses using some redundant equations. Again we chose a code of length 1000 from the ensemble $g(\lambda_1, \rho_1)$. As we discussed in the previous section, we looked for cycles of lengths 4 and 6 in the Tanner graph of the code. For any of these cycles we added the rows of H corresponding to the parity-check equations in the cycle and put these parity-check equations as rows of a matrix H'. In total we chose 374 equations. Hence, H' had 374 rows. We set $g_{max} = \infty$ and decoded 10^{10} bits. In the first experiment we used algorithm C. In the second experiment we used the same algorithm. However, we also used the parity-check equations in H' whenever the iterative decoder failed and we needed to perform the guessing procedure. Let us call

the second algorithm C′. Table 5 shows the results. In this table, g_{av} and g'_{av} are the average number of required guesses when we needed to perform the guessing process in algorithms C and C′. The table shows that the average number of guesses is substantially smaller in algorithm C′. Note also that the average number of guesses at $\epsilon = .37$ is less than one. This is because sometimes the parity-check equations in H' are sufficient to successfully finish the decoding and we do not need any guesses.

Table 5: The average number of required guesses

	g_{av}	g'_{av}
$\epsilon = .39$	2.3	1.56
$\epsilon = .38$	2.06	1.27
$\epsilon = .37$	1.74	.94

3.4 Improved Decoding Algorithms for MBIOS Channels

So far, we considered only the binary erasure channel. Our aim in this section is to generalize our improved iterative decoding algorithm to all memoryless binary-input output-symmetric (MBIOS) channels [94,100]. We use the message passing algorithm with some modifications. Let $T_A(n)$ be the average time required for the standard iterative decoding (algorithm A) of an LDPC code of length n when it is used over a MBIOS channel. Let B be an improved decoding method for the same code when used over the same channel. Let $T_B(y, n)$ be the time that algorithm B needs to decode a received word y and let $T_B(n)$ be the average time of the decoding of the code using algorithm B. We want to have:

$$T_B(n) \leq (1 + \gamma)T_A(n) \tag{46}$$

$$\forall y, \quad T_B(y, n) \leq CT_A(n) \tag{47}$$

where γ is a small constant close to zero and C is a sufficiently small constant. For the BEC, our simulations show that the algorithm we propose (algorithm C) achieves the above inequalities for $\gamma < .05$ and $C < 7$ when the length of the code is several thousands ($n \leq 5000$). For the general MBIOS channels, the proposed algorithm (algorithm D) satisfies the above inequalities for $\gamma < .05$ and $C < 40$. Thus, for the BEC both average and maximum running time are small enough. For other MBIOS channels the average

running time is still almost the same as the iterative decoding but the maximum running time can be 40 times the average running time of the standard decoding. However, for both channels, the proposed algorithms result in considerable reduction of the bit error rate with respect to the standard iterative decoding. Here, we generalize algorithm B for other MBIOS channels. It seems that generalizing algorithm C for other channels is impossible, because algorithm C takes advantage of the special structure of the BEC. In fact, that is why algorithm C is very efficient.

Suppose that an LDPC code is used for error correction over a MBIOS channel. Let $V = V(g) = \{v_1, v_2, ..., v_n\}$ and $C = C(g) = \{c_1, c_2, ..., c_m\}$ be the sets of variable and check nodes in the Tanner graph of the code, respectively. Moreover, suppose we use the standard iterative decoding (algorithm A) to decode the received words. Assume that algorithm A has small enough error probability (for example less than 10^{-2}). The iterative decoder is initialized by the log likelihood ratio (LLR) of the variable nodes based on the observation of the channel output.

As we discussed algorithms B and C in the previous section were based on the following observation. When the iterative decoding fails, knowing the values of a few bits in the stopping set is sufficient to finish the decoding successfully. We first extend this observation to arbitrary MBIOS channels. Suppose a codeword $\underline{X} = (x_1, x_2, ..., x_n)$ is transmitted over the channel, where n is the code length. Let define a function $\ell : \{-1, +1\} \longmapsto \{-\infty, +\infty\}$ as

$$\ell(x) = \begin{cases} +\infty & \text{if } x = +1 \\ -\infty & \text{if } x = -1 \end{cases} \tag{48}$$

Let $\underline{L} = (l_1, l_2, ..., l_n)$ be the (LLR) of the corresponding variable nodes $v_1, v_2, ..., v_n$ based on the observation of the channel output. Suppose the iterative decoder fails to decode a received word. In other words, after the maximum number of iterations, there are still unsatisfied check nodes in the graph. In this case there exists a set $I \subset [n] = \{1, 2, ..., n\}$ with a very small cardinality $|I|$ (usually $|I| \leq 4$) that has the following property.

Let define $\underline{L'} = (l'_1, l'_2, ..., l'_n)$ as

$$l'_i = \begin{cases} +l_i & \text{if } i \in [n] - I \\ \ell(x_i) & \text{if } i \in I \end{cases} \tag{49}$$

Now, if we initialize the iterative decoder with $\underline{L'}$, it can decode the received word correctly. We do not have proof for this statement. However, the decoding algorithm that we will introduce works well in practice.

Based on the above development, we design a decoding algorithm similar to algorithm B that was presented for the BEC in Section 3.3. Let call this algorithm D. Recall that in algorithm B, we chose the bits to be guessed from the stopping set after the decoding failure. For the general case of MBIOS channels, we need to find a method to choose the variable nodes whose values must be guessed.

We observed that the following simple method works very well in practice for choosing the variable nodes to be guessed . Let m_i be the number of unsatisfied check nodes that are adjacent to v_i when the iterative decoding fails. We choose the variable nodes that have the highest m_i's. Intuitively, by this method we find the locations of the graph for which there is a lack of information. Choosing variable nodes with high m_i, reduces the number of guesses required for successful decoding. In fact, if we just select the guessed nodes randomly, some of the guesses will not be necessary for successful decoding. Since the complexity of the algorithm increases exponentially with respect to the number of guesses, it is important to have as few guesses as possible. Note that a variable node that we choose to guess may have the correct value at the end of the iterative decoding. However, it is connected to several unsatisfied check nodes. Since we set the LLR of the guessed variable node to $+\infty$ or $-\infty$, this can help the iterative decoder to correct the values of the other bits connected to the check nodes. Again, we need to choose a maximum value for the number of guesses. Our experience shows that choosing the maximum number of guesses as five can reduce the bit error rate considerably. Note that, similar to the case of the binary erasure channel (algorithms B and C), the average running time of algorithm D, $T_D(n)$, is almost the same as algorithm A (the standard iterative decoding). However, to maintain the maximum running time small enough we need to choose a small value for the maximum

number of guesses.

We summarize algorithm D as follows. For any received codeword we perform the standard iterative decoding. If all check nodes are satisfied at the end of decoding , we are done. Otherwise, we find g variable nodes that are connected to highest number of unsatisfied check nodes at the end of the decoding and guess their values. We repeat the standard decoding but this time we initiate the algorithm with the new LLR's for the guessed bits ($+\infty$ or $-\infty$). For other bits, we use the LLR's found by the channel observation. We repeat the above procedure until either the decoding finishes successfully or all 2^g values for the guessed variable nodes are tried.

To evaluate the performance of algorithm D we chose an LDPC code of length 1000 with the following degree distribution:

$$\begin{aligned} \lambda_1(x) &= .1212x + .6364x^2 + .2424x^5, \\ \rho_1(x) &= .3818x^5 + .5939x^6 + .0243x^7. \end{aligned} \tag{50}$$

We used the expurgated ensemble. That is, we generate a code from the ensemble, and if the minimum distance of the code is small, we do not use the code and pick another code at random. Since the ensemble has asymptotically linear minimum distance [28], [83], after a few tries we will find a code with large minimum distance. We obtained the bit error rate performance for both algorithms A (the standard iterative decoding) and D. For algorithm D we chose the number of guesses $g = 5$. Figure 13 shows the performance of the decoders. We observe that algorithm D has $.35dB$ gain with respect to algorithm A at the bit error rate of 10^{-5}. The gain increases to $.5dB$ at the bit error rate of 10^{-6}. The figure also shows the performance of a randomly chosen code of length 1000 from the ensemble of (3,6) regular codes, which is known to have the best performance among the regular LDPC code ensembles. We observe that using algorithm D for decoding the above irregular code results in $1dB$ gain over the (3,6) regular code in low bit error rates.

Now we present some experimental results concerning the running time of algorithm D. We give the results for the code of length 1000 from the ensemble $g(\lambda_1, \rho_1)$. We decoded 10^9 bits over the binary-input additive white Gaussian noise (BIAWGN) channel and measured the average and the maximum running time for the decoding of received blocks. Let $T_A(n)$

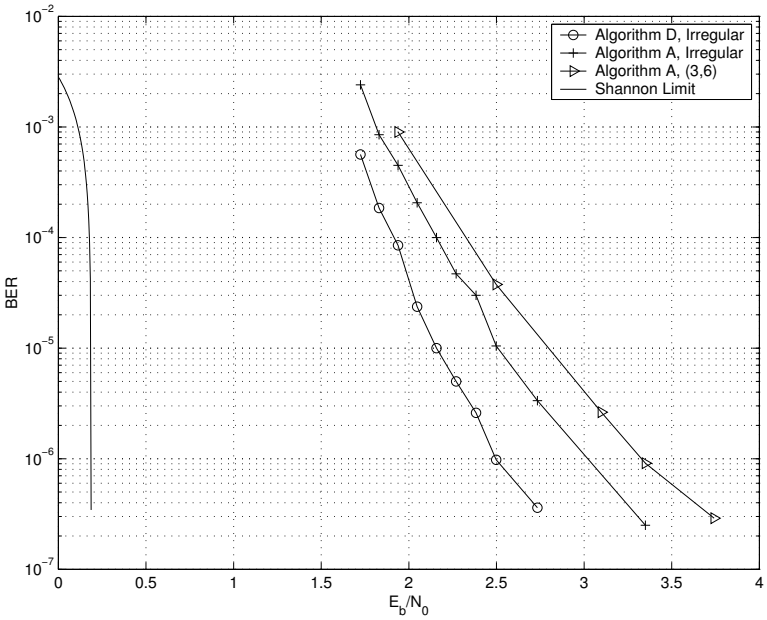

Figure 13: Comparisons of the bit error rates of algorithms A and D for an irregular code of length $n = 10^3$ and the (3,6) regular code decoded by algorithm A over the BIAWGN channel.

and $T_D(n)$ show the average time of decoding of the LDPC code of length n from the given ensemble using algorithms A and D, respectively. Our simulations suggest that the estimate of $\frac{T_D(n)}{T_A(n)}$ is equal to 1.12. This implies that the average running time of algorithms A and D are almost the same. Recall that we defined $T_D(y, n)$ as the time that algorithm D needs to decode a received word y. Let $T_D^m(n)$ denote the maximum value of $T_D(y, n)$ over all the received blocks. We define $R_D(n) = \frac{T_D^m(n)}{T_A(n)}$. Our simulations show that $R_D(n) = 38.4$, approximately. In other words, in the worst case the time that Algorithm D needs to decode a received word can be 38.4 times the average running time of algorithm A. In [101] we have estimated that $R_C(n) = 6.7$, for the BEC and a half-rate code with length 1000.

3.5 Improved Decoding for Non-Uniform Channels

In Chapter 5 we will investigate the application of LDPC codes for non-uniform channels [104] and [106]. We will introduce a scheme for designing LDPC codes over these channels. We also show that the punctured codes can be viewed as a special case of non-uniform channels. In this section we study some properties of LDPC codes on non-uniform channels and show that the improved decoding algorithm proposed in this chapter is more effective

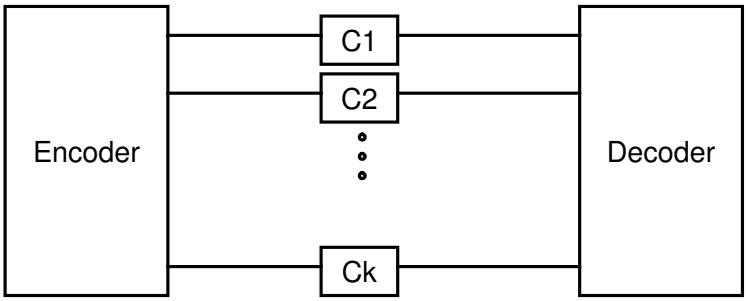

Figure 14: Several parallel channels.

on these channels if used properly. Specifically, we study the effect of the algorithm on punctured LDPC codes.

A nonuniform channel can be considered as several parallel independent subchannels as it is shown in Figure 14. We assume that we use one LDPC code over the set of subchannels. Thus, different bits in a codeword may be transmitted over different subchannels. Some practical examples of non-uniform channels are volume holographic memory (VHM) systems, orthogonal division frequency multiplexing (OFDM) systems, and multilevel coding.

In VHM systems, the information is recorded and retrieved in the form of two-dimensional data pages, i.e, two-dimensional patterns of bits. These bits are subject to different sources of noise. The SNR decreases as we move from the center to the corner of the page. Typically, raw bit error rate might vary by two or three orders of magnitude over a page. As we explained in [106] we can divide a VHM page to k regions R_i such that bits of the same region have almost the same raw error probability. Any region in the page corresponds to one of the subchannels in Figure 14. A similar situation exists in OFDM systems that consist of several parallel channels with different SNR's.

Suppose we use a code of length n. We transmit every codeword such that $n^{(j)}$ bits from each codeword are transmitted over the j'th channel. In [106] we defined an ensemble $g(\Lambda, \rho)$ of LDPC codes. We showed that they have some good properties. For convenience, we repeat the definition of the ensemble here. The main point is that in the ensemble $g(\Lambda, \rho)$, bits of different types may have different degree distributions. Formally, let $(x_1, x_2, ..., x_n)$ be a codeword. Let also $W^{(j)}$ be the set of bits from the codeword that are transmitted

over the j'th channel (type j bits). Thus we have $|W^{(j)}| = n^{(j)}$, where $|\cdot|$ denotes the cardinality of the set. For example, in the VHM system, $W^{(j)}$ is the set of bits in the j^{th} region (i.e., $W^{(j)} = \{x_i : x_i \in R_j\}$). Now we define the ensemble $g(\Lambda, \rho)$ of bipartite graphs that we propose for nonuniform error protection. Let E be the set of edges in the graph and let $E^{(j)}$ be the set of edges that are incident with a variable node of type j. Also let $E_i^{(j)}$ be the set of edges adjacent to the variable nodes of type j and degree i. We define

$$\lambda^{(j)}(x) = \sum \lambda_i^{(j)} x^{i-1} \tag{51}$$

where

$$\lambda_i^{(j)} = \frac{|E_i^{(j)}|}{|E^{(j)}|} \tag{52}$$

Let $\Lambda = \{\lambda^{(j)}(x) : j = 1, ..., k_r\}$. Let also $\rho(x) = \sum \rho_i x^{i-1}$ be as defined in [54]. We define the ensemble $g(\Lambda, \rho)$ as the ensemble of bipartite graphs with the degree distributions given by Λ and ρ.

Now we give some properties of the densities of the messages in the belief propagation algorithms on the ensemble $g(\Lambda, \rho)$. These properties are specifically useful for applying the improved decoding algorithm. Let $m_{vc}^{(l),(j)}$ denote the message that is sent from a variable node of type j (i.e. $v \in W^{(j)}$) to its incident check node c in the l'th iteration of the message passing algorithm. Let also $m_{cv}^{(l)}$ denote the message that the check node c sends to its incident variable node. Let $F_l^{(j)}$ and Q_l denote the average asymptotic distributions of random variables $m_{vc}^{(j),(l)}$ and $m_{cv}^{(l)}$, respectively.

Consider an MBIOS channel with parameter θ, where $\theta \in [\theta_{min}, \theta_{max}]$ and $\theta_{min}, \theta_{max} \in \mathbb{R} \cup \{-\infty, +\infty\}$. For example, for the BIAWGN channel, θ can be considered as the variance σ of the noise. Let $\mathcal{C}$ be a class of channels with parameter θ. Thus, any channel C_θ in $\mathcal{C}$ in is uniquely determined by its variable θ. A channel in $\mathcal{C}$ with parameter θ_0 is called C_{θ_0}. The capacity of the channel C_{θ_0} is shown by c_{θ_0}. Similar to [112], we consider physically degraded channels. For clarity of exposition we assume that if $\theta_1 < \theta_2$, then C_{θ_2} is physically degraded with respect to C_{θ_1}.

Consider the case that in Figure 14 all subchannels are the same type but have different channel parameter. Moreover, all subchannels belong to a class of physically degraded

channels $\mathcal{C}$ as explained above. Suppose we use an LDPC code from the ensemble $g(\Lambda, \rho)$ over these channels. Assume the variable nodes of type j are transmitted through the channel C_{θ_j}. Then we have the following theorem [100]:

Theorem 6. *If $\theta_i < \theta_j$ and $\lambda^{(i)}(x) \equiv \lambda^{(j)}(x)$, then for any l we have $P_e(F_l^{(i)}) \leq P_e(F_l^{(j)})$.*

Proof. This theorem can be proved using similar discussions as in [112]. Let u_i be a variable node of type i in the Tanner graph of the code. Thus, u_i receives observation from the output of the channel C_{θ_i}. Let

$$B_l = \bigcup_{k=0}^{2l} N^k(u_i) \tag{53}$$

be the neighborhood of u_i of depth l. Let g_B be the graph induced by the vertices in B_l. In the graph g_B, the variable node u_i receives information from the channel C_{θ_i}, and other variable nodes receive observation from possibly different channels. Let O_i be the observation of u_i and let O be the set of observations of the other variable nodes in B_l. Now, assuming that g_B is a tree, $P_e(F_l^{(i)})$ is the error probability of the ML decoder based on the observations of the variable nodes in B_l (i.e., $O \cup \{O_i\}$). Now, in the above graph, if we just replace the variable node u_i with the variable node u_j that receives observation from the channel C_{θ_j}, again $P_e(F_l^{(j)})$ is the error probability of the ML decoder based on the observations of the variable nodes in B_l (i.e., $O \cup \{O_j\}$). Since C_{θ_j} is physically degraded with respect to C_{θ_i}, we can consider O_j as the result of passing O_i through another channel C'. Since given O_i, the observation O_j is independent of the value of the transmitted bit, $P_e(F_l^{(i)})$ is the error probability of the ML decoder based on the observations $O \cup \{O_i\} \cup \{O_j\}$. Thus $P_e(F_l^{(i)}) \leq P_e(F_l^{(j)})$ for the given g_B. Since $\lambda^{(i)}(x) \equiv \lambda^{(j)}(x)$, any structure of the neighborhood (the graph g_B and the channels from which the bits receive information) has the same probability of occurrence for u_i and u_j. Thus we conclude that $P_e(F_l^{(i)}) \leq P_e(F_l^{(j)})$. □

Theorem 6 states that, under certain conditions, the bits that have higher error probability before the decoding, have higher error probability after the decoding as well. We have the following corollary.

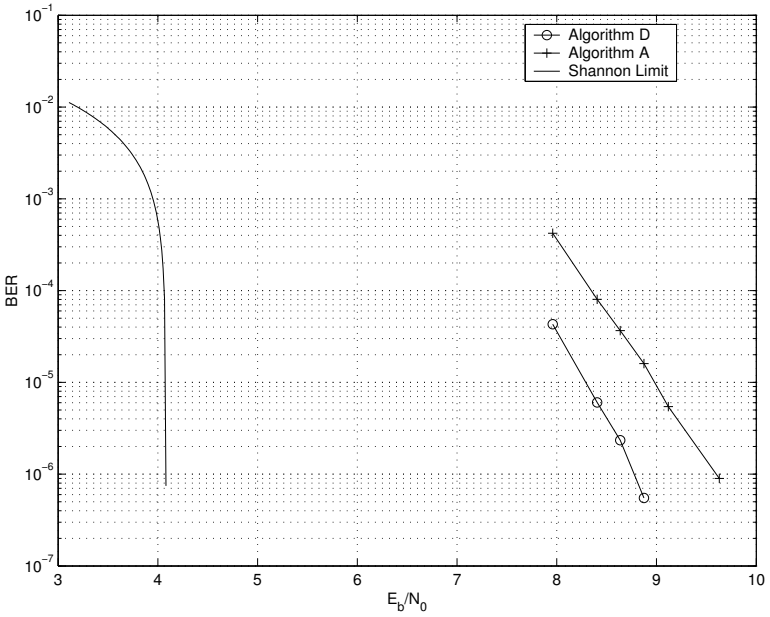

Figure 15: Comparisons of the bit error rates of algorithms A and D for the (3,6)-regular LDPC code over the BIAWGN channel.

Corollary 3. *In a regular ensemble, the variable nodes that receive information from channels with smaller capacity have higher error probability after the decoding.*

In the following, we study applications of improved decoding (algorithm D) on non-uniform channels. In the guessing process we choose a variable node to be guessed. As we mentioned, it is better to choose the variable node from the parts in the graph that there is a lack of information. Since the likelihood of the error for the nodes that receive information from the channels with smaller capacity is higher, one simple method in the guessing process is to give priority to these nodes. Fortunately, our simulations show that this simple method works very well and considerably improves the decoding performance.

Punctured codes can be considered as a special case of non-uniform channels in which the punctured bits are transmitted through a channel with zero capacity. In the next section we study this special case of non-uniform channels more deeply. To observe the performance of algorithm D on non-uniform channels, we chose a $(3, 6)$ regular LDPC code of length 1000. We randomly chose 37.5 percent of the variable nodes (i.e, 375 variable nodes) and designate them as punctured variable nodes. Thus the resulting code has the rate 0.8. We then evaluate the performance of algorithms A and D for this code. As discussed above,

in choosing the variable nodes to guess, the punctured bits were given priority. Figure 15 shows the performance of the algorithms. We observe that improvements close to $1dB$ is gained at low bit error rates using algorithm D.

3.6 Application of Pseudo-Codewords to the Analysis of Algorithm D

Simulation results given in previous sections suggest that we can considerably improve the performance of LDPC codes using algorithm D. However, so far our results are totally based on simulations. In this section, we provide some discussions based on pseudo-codewords [36, 37, 62] that gives us some insight into algorithm D. We first briefly provide some definitions and results from [62]. For more details, readers are referred to [62]. Let $G = (V, E)$ be a graph with vertex set $V = V(G) = \{v_1, v_2, ..., v_l\}$ and edge set E. Finite covers are defined in [62] in the following way.

Definition 1. *A finite degree m cover of $G = (V, E)$ is a graph $\hat{G}$ with vertex set $\hat{V} = \bigcup_{i=1}^{l} \hat{V}_i$, where each set $\hat{V}_i = \{\hat{v}_{i,1}, \hat{v}_{i,2}, ..., \hat{v}_{i,m}\}$ contains m vertices. The edge set $\hat{E}$ of $\hat{G}$ is chosen as a subset of $\{\{\hat{v}_{i,s}, \hat{v}_{j,r}\} : \{v_i, v_j\} \in E, s, r \in \{1, 2, ..., m\}\}$ such that for each vertex $\hat{v}_{i,s} \in \hat{V}$, we have $deg_{\hat{G}}(\hat{v}_{i,s}) = deg_G(v_i)$, and $|N(\hat{v}_{i,s})| = |N(v_i)|$. Moreover, $N(\hat{v}_{i,s})$ contains exactly one vertex $\hat{v}_{j,r}$ for all j for which $v_j \in N(v_i)$.*

Loosely speaking, the graph $\hat{G}$ is obtained by replicating every vertex in G m times and introducing edges such that the local adjacency relationships between replicated nodes are preserved.

Let g be a Tanner graph of an LDPC code C. Let also $V = V(g) = \{v_1, v_2, ..., v_n\}$ and $C = C(g) = \{c_1, c_2, ..., c_m\}$ be the sets of variable and check nodes in the Tanner graph of the code, respectively. For simplicity, suppose we use the code C over a BIAWGN channel. Let $\hat{g}$ be a finite m cover of g and $\hat{C}$ be the corresponding code. For any codeword $\hat{c} \in \hat{C}$ a vector $w = w(\hat{c})$ is defined in [62], which is called a pseudo-codeword of C. The fundamental cone of the graph g, denoted by $\mathcal{F}(g)$, is defined in [62] and is related to the set of pseudo-codewords $w(\hat{c}) = w$ taken over all covers of g of all degrees $m = 1, 2, ...$

Let $\eta = (\eta_1, \eta_2, ..., \eta_n)$, be the set of received LLR's from the channel. Assuming that

all-one word was transmitted, it is shown in [62] that the decoding is successful if and only if for all $w \in \mathcal{F}(g)$, we have $\langle w, \eta \rangle > 0$.

Using the concept of pseudo-codewords we now determine a necessary and sufficient condition for the success of algorithm D [100]. Assume $\eta = (\eta_1, \eta_2, ..., \eta_n)$ be the LLR vector received from a BIAWGN channel. Moreover, assume that the standard iterative decoding has failed. Define

$$\mathfrak{P}(\eta) = \{w \in \mathcal{F}(g) : \langle w, \eta \rangle < 0\}. \tag{54}$$

The following lemmas give necessary and sufficient conditions for the success of algorithm D. Let $\mathcal{G} = \{i : v_i \text{ is guessed}\}$ be the set of indices of the guessed variable nodes in algorithm D. Let η' be the new LLR's imposed by algorithm D (i.e., by replacing the LLR of the guessed bits by $-\infty$ or $+\infty$).

Lemma 5. *algorithm D succeeds only if for all $w \in \mathfrak{P}(\eta)$, there exists $i \in \mathcal{G}$, such that $w_i > 0$.*

Proof. Suppose there exists $w^* \in \mathfrak{P}(\eta)$, such that for all $i \in \mathcal{G}$, we have $w_i^* = 0$. Then, $\langle w^*, \eta' \rangle = \langle w^*, \eta \rangle < 0$. Thus the iterative decoding fails independent of the values of the guessed bits. □

We now show that the condition of Lemma 6, is actually a sufficient condition for algorithm D to converge to the ML decoding.

Lemma 6. *If for all $w \in \mathfrak{P}(\eta)$, there exists $i \in \mathcal{G}$, such that $w_i > 0$, then algorithm D can find the ML decoded codeword.*

Proof. Assume all-one codeword is transmitted. Since we check all possible values for the guessed bits, at some point we will guess the correct value for all the guessed bits. That is the LLRs for all the guessed bits become $+\infty$. Since for all $w \in \mathfrak{P}(\eta)$, there exists $i \in \mathcal{G}$, such that $w_i > 0$, we conclude that for all $w \in \mathfrak{P}(\eta)$, $\langle w, \eta' \rangle = +\infty$. Thus the decoding is successful.

Therefore, when the guesses are correct the decoding is successful. On the other hand, it is possible that the algorithm converges to a wrong codeword for some other guessed

values. In fact, for good LDPC codes, this is very unlikely. However, in these situations, at the end of decoding we get more than one decoded valid codewords. Since the number of these codewords is small, by a simple ML test, weobtain the result of ML decoding. □

Using the above result one may try to analyze the performance of algorithm D. In fact, this is one of the future direction of our research.

3.7 Stopping Sets

As we have already seen, stopping sets determine the performance of LDPC codes over the BEC. In fact, as we will see later, stopping sets play an important role for other MBIOS channels also. In this section we study stopping sets further. The stopping set distributions of LDPC codes has been studied in [19,83], and [109]. Not only, do stopping sets determine the performance of the iterative decoding, but also they are useful in the study of the LDPC codes on other channels [122], [96]. A fundamental problem is to determine if a given bipartite graph has a stopping set of a given size. We refer to this problem as stopping sets sizes (SS). This problem is important from several points of view. For example, if SS$\in$ P, then we could find the size of the minimum stopping set in a graph and determine its erasure correction radius in polynomial time. Moreover, this could be used in removal of some problematic stopping sets. On the other hand, the NP-hardness of SS implies that we cannot find a deterministic polynomial time algorithm to obtain the stopping set distribution of LDPC codes, which is indeed an important characteristic of the code. The same is true about the distance distribution of the code . This amplifies the importance of the average distribution analysis such as those given in [83], [19], [109] and [66]. It also encourages to study the concentration of these distributions on their average. In this section, we show that SS is NP-hard. Some fundamental problems in coding have been shown to be intractable. For example, see [9] and [123]. It is also easy to see that most of these problems remain intractable even if the code has a sparse parity check matrix. This is because, the codes that are used for the proofs of the theorems have sparse parity check matrices.

3.7.1 Intractability of SS

Let $H = [h_{ij}]$ be an m by n sparse binary matrix. In other words, there exists a constant number d such that

$$\frac{\|H\|}{n} \leq d \tag{55}$$

where $\|H\|$ is the number of nonzero elements in H. The Tanner graph [121] of the matrix H, $G(H)$, is a bipartite graph with bipartition $V(G)$ and $C(G)$, where $V(G) = \{v_1, v_2, ..., v_n\}$ and $C(G) = \{c_1, c_2, ..., c_m\}$ are the sets of variable and check nodes, respectively. The nodes c_i and v_j are adjacent if h_{ij} is equal to 1. A stopping set S is defined as a subset of $V(G)$ such that all neighbors of S are connected to S at least twice [27].

We will show that SS is NP-hard [97]. The proof uses similar ideas to [9]. We show the intractability by reducing any instance of the 3-dimensional matching (3DM) problem to an instance of SS. We note that 3DM is NP-complete and it can be stated as follows [40].

3-DIMENTIONAL MATCHING(3DM)

INSTANCE: A set $L \subseteq W \times W \times W$ where W is a set having q elements.

QUESTION: Does L contain a matching, that is, a subset $L' \subseteq L$ such that $|L'| = q$ and no two elements of L' agree in any coordinate?

Let us first give some definitions. We show vectors by $1 \times K$ matrices. For two binary matrices A and B, $A \times_b B$ is multiplication of the two matrices over $GF(2)$, while $A \times_{\mathbb{R}} B$ denotes their multiplication over the real field. For $1 \leq i < j \leq m$ and a vector $Y = (y_1, y_2, ..., y_m)$, we define $Y([i:j])$ as $Y([i:j]) = (y_i, y_{i+1}, ..., y_j)$. Let $[q] = \{1, 2, ...q\}$ and $L = \{a_1, a_2, ..., a_l\} \subseteq [q] \times [q] \times [q]$, where $a_i = (a_{i1}, a_{i2}, a_{i3})$. We define the $|L| \times 3q$ matrix $B(L)$ as follows. Let $l = |L|$, then $B(L) = [B_1, B_2, B_3]$, where $B_i = [b_{kp}^{(i)}]$ is an l by q binary matrix and $b_{kp}^{(i)} = 1$ if and only if $a_{ki} = p$. If r_j shows the j'th row of $B(L)$, then L contains a matching if and only if there is a set $J \subseteq \{1, 2, ..., l\}$ with $|J| = q$ such that

$$\sum_{j \in J} r_j = (1, 1, ..., 1) \tag{56}$$

In the above sum, r_j's are added together by XORing their coordinates. For the above set L we construct the parity check matrix $H(L)$ as Figure 16 in which $B'(L)$ is the matrix

obtained by transposing $B(L)$ and I_t is the $t \times t$ identity matrix for a positive integer t. A binary vector $X = (x_1, x_2, ..., x_n)$ is said to be a stopping vector of an m by n sparse binary matrix H if the vector Y defined by

$$Y = (y_1, y_2, ..., y_m) = X \times_{\mathbb{R}} H' \tag{57}$$

satisfies $y_j \neq 1$, for $j = 1, 2, ..., m$. Any stopping vector x is associated with a stopping set in the corresponding Tanner graph.

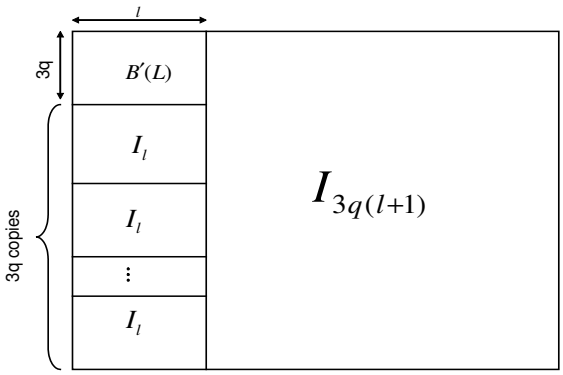

Figure 16: Structure of the matrix $H(L)$.

Lemma 7. *A set $L = \{a_1, a_2, ..., a_l\} \subseteq [q] \times [q] \times [q]$ contains a matching if and only if $H(L)$ has a stopping vector of weight $w = \|X\| = 3q^2 + 4q$.*

Proof. Suppose the matrix $H(L)$ has a stopping vector X with $w = \|X\| = 3q^2 + 4q$. Let $X = (x_1, x_2, ..., x_{3q(l+1)+l})$. We write $X = (X_1, X_2)$ where $X_1 = (x_1, x_2, ..., x_l)$ and $X_2 = (x_{l+1}, x_2, ..., x_{3q(l+1)+l})$. Then, we obtain

$$Y = (y_1, y_2, ..., y_{3q(l+1)}) = X \times_{\mathbb{R}} H' = (X_1 \times_{\mathbb{R}} B(L), X_1, X_1, ..., X_1) + X_2 \tag{58}$$

By the assumption we have $y_j \neq 1$, for $j = 1, 2, ..., 3q(l+1)$. Then $X_2 = (X_3, X_1, X_1, ..., X_1)$ where $X_3 = (x_{l+1}, x_2, ..., x_{3q+l})$. Thus $X = (X_1, X_3, X_1, X_1, ..., X_1)$ and we have

$$\begin{aligned} \|X\| = \|X_1\| + \|X_2\| = \\ \|X_1\| + \|X_3\| + 3q\|X_1\| = \\ (3q+1)\|X_1\| + \|X_3\| \end{aligned} \tag{59}$$

On the other hand, we have

$$\|X\| = 3q^2 + 4q = (3q+1)q + 3q. \tag{60}$$

Since the length of the vector X_3 is equal to $3q$, we have $\|X_3\| < 3q + 1$. Therefore, combining (59) and (60) we obtain

$$\begin{aligned} \|X_1\| &= q \\ \|X_3\| &= 3q \end{aligned} \tag{61}$$

Thus we have $X_3 = (1, 1, ..., 1)$. Let $Z = (z_1, z_2, ..., z_{3q}) = X_1 \times_{\mathbb{R}} B'(L)$. As we showed perviously

$$\begin{aligned} y([1:3q]) = (y_1, y_2, ..., y_{3q}) = \\ Z + X_2([1:3q]) = \\ Z + (1, 1, ..., 1) \end{aligned} \tag{62}$$

Since $y_j \neq 1$, we have $z_j \geq 1$ for $j = 1, 2, ..., 3q$. In fact, we claim that $z_j = 1$ for $j = 1, 2, ..., 3q$. This is because $\|X_1\| = q$, and thus the vector $Z = X_1 \times_{\mathbb{R}} B'(L)$ is constructed by adding q rows of $B(L)$. But each row in $B(L)$ has exactly three 1's. Thus $\|Z\| = 3q$. Combining this with $z_j \geq 1$ for $j = 1, 2, ..., 3q$ results in $z_j = 1$ for $j = 1, 2, ..., 3q$. Recalling (56), we conclude that the set L contains a matching.

Conversely, suppose the set $L = \{a_1, a_2, ..., a_l\} \subseteq [q] \times [q] \times [q]$ contains a matching . Then we can find a vector X_1 of length l and weight q, such that $X_1 \times_{\mathbb{R}} B'(L) = (1, 1, ..., 1)$. If we choose a $1 \times 3q$ vector $X_3 = (1, 1, ..., 1)$, then $X = (X_1, X_3, X_1, X_1, ..., X_1)$ is a stopping vector of $H(L)$ of weight $w = \|X\| = 3q^2 + 4q$.

□

We now show that SS is NP-hard. The SS is defined formally as follows.

STOPPING SETS SIZES(SS)

INSTANCE: A sparse binary matrix H and a finite number d such that $\frac{\|H\|}{n} \leq d$ and an integer w.

QUESTION: Does H have a stopping vector of weight w?

Theorem 7. *SS is NP-hard [97].*

Proof. We show that any instance of 3DM can be reduced to an instance of SS. Suppose we are given a set $L \subseteq W \times W \times W$ where $|L| = l$ and W is a set having q elements. We can construct the $3q(l+1) \times (3q(l+1)+l)$ matrix $H(L)$ as figure 16 in polynomial time. Let $N = 3q(l+1)+l$. The matrix H is sparse for any input set L, because

$$\frac{\|H\|}{N} = \frac{3ql + 3l + 3q(l+1)}{3q(l+1)+l} = \\ 1 + \frac{3ql + 2l}{3q(l+1)+l} \leq 1 + 2 = 3. \tag{63}$$

Thus by Lemma 7, the 3DM problem reduces to an SS problem for which we have $H = H(L)$, $w = 3q^2 + 4q$, and $d = 3$. □

Discussion: The above theorem shows that in general the problem of determining if a sparse graph has a stopping set of a given size is NP-hard. However, in some practical situations this does not impose a problem. For example, we may want to check if a given bipartite graph has a stopping set of size less than a constant number c, which does not depend on the size of the graph. This task can be done in $O(n^c)$ time. As another example, it is known that the error floor problem is caused by the stopping sets that contain only variable nodes of degree two. Let us call these stopping sets, D-2 stopping sets. If $\lambda_2\rho'(1) > 1$, then with high probability, the graph contains small D-2 stopping sets that dominate the error floor effect [109], [83]. Thus, in this situations when we generate a random graph from the ensemble, we know that the code will suffer from error floor problem. There are several methods to deal with this problem. For example see [122].

D-2 stopping sets consist of several variable-disjoint cycles(by variable-disjoint cycles we mean the cycles whose sets of variable nodes are disjoint). Thus, the small D-2 stopping sets can be found efficiently using standard graph algorithms. Another method to find D-2 stopping sets is to take a codeword and erase the bits corresponding to degree-two variable nodes. Then perform the iterative decoding on the resulting word. In fact, for the codes with $\lambda_2\rho'(1) < 1$, we can usually avoid D-2 stopping sets as implied by the following theorem [97].

Theorem 8. *Consider the ensemble (λ, ρ) of LDPC codes that satisfy $\lambda_2\rho'(1) < 1$. Then we have the following:*

(a) *If the equation $f(x) = 1 - \rho(1 - \lambda_2 x) = x$ does not have any solution in $(0, 1]$, there exists a strictly positive constant δ such that with probability at least δ, a randomly chosen graph from the ensemble does not have any D-2 stopping set.*

(b) *If $f(x) = x$ has a solution in $(0, 1]$, then with high probability the graph has D-2 stopping sets.*

Proof. Since $\lambda_2\rho'(1) < 1$, with strictly positive probability we do not have sublinear D-2 stopping sets [83], [109]. The condition $f(x) < x$ for $(0, 1]$ asserts that with high probability we do not have linear-size D-2 stopping sets. Because the iterative decoding is successful with high probability if only the degree-two variable nodes are removed from the graph (This is a special case of non-uniform channels for which we can find the code performance using the methods described in [106]). On the other hand if $f(x) = x$ for some $x \in (0, 1]$, then the decoding fails with high probability, and hence there are 2-D stopping sets in the graph. □

Thus for the ensembles satisfying the condition (a) of the theorem, we can simply generate a random graph from the ensemble, and then by performing the iterative decoding we can check if there exists a 2-D stopping set. If yes, we can repeat the process. After a few iterations we obtain a graph with no 2-D stopping sets.

It is worth nothing that the condition (a) of Theorem 8 usually holds for practical ensembles. For example, here is an evidence of this statement.

Lemma 8. *For an ensemble satisfying $\lambda_2\rho'(1) < 1$, we have $\lambda'_2 < 1 - R$, where λ'_2 is the fraction of degree-two variable nodes and R is the code rate.*

This lemma is proved in Appendix A. By Lemma 8 we have $\lambda'_2 < 1 - R$, thus if the threshold of the code, ϵ_{th}, is sufficiently close to $1-R$ (which is usually the case for optimized codes), then we have $\lambda'_2 < \epsilon_{th}$. Thus, it is reasonable to expect that the decoder can recover almost all variable nodes of degree two when other variable nodes are known. If this holds,

then the code does not have linear-size 2-D stopping sets. However, for a proof of this for a specific ensemble, we need to check if the conditions of Theorem 8 holds.

3.8 Conclusion

In this chapter we studied some properties of the ML and iterative decoding of LDPC codes. We derived both lower and upper bounds on the ML capacity of the ensembles of LDPC codes on the BEC. The tightness of the bounds was depicted for regular codes by using some examples. We proposed two algorithms for decoding LDPC codes over the BEC. It was shown by simulations that the proposed algorithms has a better bit error rate than the standard iterative decoder. More specifically, the improvement up to three orders of magnitude was obtained in the bit error rate. It was also demonstrated that the proposed algorithm (algorithm C) has almost the same running time as the iterative decoder. Therefore, we conclude that the algorithm can be considered as an efficient method for decoding LDPC codes over the BEC. Since finding good finite-length LDPC codes is still a challenging problem, the decoding scheme presented in this chapter may compensate for this problem.

We then generalized the improved decoding algorithms to all MBIOS channels. The algorithm has considerably smaller bit error rates than the standard iterative decoding algorithm. For the BEC, both the average running time and the maximum running time of the proposed algorithm (algorithm C) is small. For other MBIOS channels, the average running time of the proposed algorithm (algorithm D) is almost the same as the standard iterative decoder. However, the maximum running time of the algorithm can be as large as 40 times the average running time of the iterative decoder. We showed that if the algorithm applied properly, it can be more effective on non-uniform channels.

Finally, we showed that the problem of determining weather a sparse graph has a stopping set of a given size is NP-hard. However, we pointed out that in certain practical situations this does not impose any difficulty in the design and performance estimation of the code.

CHAPTER IV

PERFORMANCE OF LDPC CODES WITH LINEAR MINIMUM DISTANCE

4.1 Introduction

In some applications, it is necessary to design codes that do not suffer from the error floor problem at the desired bit error rates (BERs) and at the same time have rates close to the channel capacity. For example, in some page-oriented memories, LDPC codes can result in very efficient coding schemes [107]. In these memories, we can use large block lengths and thus we get performance close to the Shannon limit. However, BERs less than 10^{-12} are required. Since the storage capacity of the system is directly proportional to the code rate, it is very important that the code rate be close to the capacity of the channel. Thus we need to design LDPC codes that do not show error floor for the BERs higher than 10^{-12} and at the same time have a threshold near the Shannon limit.

One method to solve the error floor problem is to use an outer code. In this method we use the outer code to reduce the BER. This method slightly increases the complexity of the system. This is specifically undesirable in page-oriented memories where simple and fast decoding algorithms are required. Moreover, using an outer code results in rate loss; however, the rate loss is usually small. There are also methods for decreasing the error-floor effect for the capacity-approaching codes [122]; however, these methods are sometimes not effective for the BERs required by storage systems. Depending on the application, these methods may or may not be suitable. As it will be described in more detail, an alternative option is to use codes with linear minimum distance. These codes have some desirable properties other than good error floor performance. Thus, in this chapter our aim is to study codes with linear minimum distance and to find bounds on their achievable rates. These results are useful for choosing codes for a given system.

As we stated in Chapter 2, it has been shown in [112] and [54] that any ensemble (λ, ρ) has a threshold under the iterative decoding. If the noise level of the channel is bellow the threshold, the BER of the iterative decoder tends to zero as the code block length tends to infinity. On the other hand, if the noise level is above the threshold, the BER is bounded away from zero. Throughout the chapter, by threshold we mean the the threshold of the code ensemble under the iterative decoding.

The error floor problem is related to the minimum distance and the minimum stopping set size of the code. As it is shown in [28], [27] , and [83], a suitably expurgated ensemble (λ, ρ) of LDPC codes has a linear typical minimum distance and minimum stopping set size if $\lambda'(0)\rho'(1) < 1$. Here, a constant fraction of the codes in the ensemble with low minimum stopping set size are removed in the expurgation. On the other hand, if $\lambda'(0)\rho'(1) > 1$ the size of the minimum stopping set and the minimum distance is sublinear with high probability. The codes with small minimum distance and small minimum stopping set (the ones with $\lambda'(0)\rho'(1) > 1$) suffer from the error floor problem. On the other hand, if the minimum distance is linear, the error-floor effect is reduced substantially. For the binary erasure channel (BEC) with low enough channel erasure probability, using a simple union bound we can show that the BER of an expurgated ensemble with $\lambda'(0)\rho'(1) < 1$ decreases exponentially with respect to the code length [101], [19]. Thus the code shows lower error floor effect for the corresponding erasure probability range. Although this has not been shown for other channels, simulations clearly show the superiority of these codes in terms of the error-floor effect over the codes having a sublinear minimum distance. It is shown in [120] that (assuming that the first two derivatives of $1 - \rho^{-1}(1 - x)$ are positive in $(0, 1)$) capacity achieving LDPC codes over the BEC satisfy $\lambda'(0)\rho'(1) > 1$ and hence have sub-linear minimum distance. Thus, they are very likely to suffer from the error-floor problem. Code ensembles satisfying $\lambda'(0)\rho'(1) < 1$ present other good properties such as having a strictly positive relative erasure correction radius. In other words, if the size of the minimum stopping set is greater than δn, where n is the code length and δ is a positive constant, then the code is guaranteed to recover all the erased bits provided that the number of erased bits is less than or equal to δn. Their iterative decoding is also faster than the capacity

achieving ones [120].

The question that arises here is how close we can get to the Shannon limit while the minimum distance is maintained linear with respect to the code length. In other words, how much do we possibly lose by restricting to codes with linear minimum distance? Here we are concerned with this question.

In this chapter, we first study small cycles and stopping sets. We give probability distributions of these small subgraphs. Using the results we find the error floor that is caused by these small subgraphs. We find lower bounds on the achievable rates over memoryless binary-input output-symmetric (MBIOS) channels using LDPC codes with linear minimum distance when decoded using the belief propagation algorithm. Then, we obtain upper bounds for the rate of these ensembles over the BEC. We give upper bounds similar to [8] for codes with a given right degree distribution. We will compare these bounds with the ones given in [8] to estimate the rate loss due to restricting to codes with the linear minimum distance property. We think that, like almost any other properties of LDPC codes, the study of this question over the BEC can provides better understanding of the problem over other channels.

Note that here we are concerned with error floor for relatively large block lengths (e.g., $n > 5000$). The error floor for short codes is more complicated [110], [111]. For simplicity, we sometimes give our results for right-regular ensembles; however, the results can be trivially generalized for other ensembles.

Throughout the chapter we assume the following terminology. By a graph we mean a simple graph, i.e., a graph with no loops (edges joining a vertex to itself) and no multiple edges (several edges joining the same two vertices). However, a multigraph may have loops or multiple edges. Let A be a subset of the vertices in the graph g. $N(A)$ shows the set of neighbors of A in g. Let D be a subgraph of g such that its vertex set is A. We say D is induced by A if D contains all edges of g that join two vertices in A. For a graph g, $deg_g(v)$ is the degree of v in g. If V is the set of vertices in g and $U \subseteq V$, then $deg_U(v)$ is the number of neighbors of v in U. For a random variable X, we show its distribution by $F_X(x)$. If the random variable is absolutely continuous, we represent its density function by $f_X(x)$.

Similar to [54], we define $P_e(F_X)$ =Pr$\{X < 0\} + \frac{1}{2}$Pr$\{X = 0\}$. For a random variable Y, $E(Y)_k$ shows the k'th factorial moment. That is $E(Y)_k = E[Y(Y-1)...(Y-k+1)]$. Let ε_n be an event depending on a parameter n. We say that ε_n holds asymptotically almost surely if Pr$\{\varepsilon_n\}$ tends to 1 as $n \to \infty$. $Po(\eta)$ is used for random variables with poisson distribution and average η.

4.2 Distributions of Small Cycles and Stopping Sets

In this section we find the asymptotic probability distribution of small cycles and stopping sets in ensembles of LDPC codes. These results can be proved using similar arguments used for finding distributions of short cycles in random regular graphs [10, 53]. A stopping set S is defined in [27] as a subset of V, the set of variable nodes, such that all neighbors of S are connected to S at least twice. Let ε be the subset of the set of variable nodes that is erased by the channel. It is proved in [27] that the iterative decoding fails if and only if ε contains a stopping set. In [27] it is also shown that the set of remaining erasures when the decoder stops is equal to the unique maximal stopping set of ε.

Let $g(n, \lambda, \rho)$ be the ensembles of multigraphs of LDPC codes of length n with degree distribution given by $\lambda(x)$ and $\rho(x)$ ($\lambda(x) = \sum_{i\geq 2} \lambda_i x^{i-1}, \rho(x) = \sum_{i\geq 2} \rho_i x^{i-1}$) and similarly let $g(n, d_v, d_c)$ be the ensembles of bi-regular multigraphs of LDPC codes of length n with variable node degree d_v and check node degree d_c [54].

Here we use the same ensemble described in [112]. That is, for example to generate a code from the $g(n, d_v, d_c)$ ensemble we do the following. To each variable or check node we assign d_v or d_c sockets respectively. We label the variable nodes and check nodes separately with the set $\{1, 2, ..., nd_v\}$. We then pick a random permutation π on $E = nd_v$ letters. For each i, we put an edge between the socket i and $\pi(i)$. Let $X_l(n, d_v, d_c)$ be the number of cycles of length $2l$ in $g(n, d_v, d_c)$. Then we have the following theorem [96].

Theorem 9. *Let* $\eta_l = \frac{\left[(d_v-1)(d_c-1)\right]^l}{2l}$ *and let* $X_{l\infty} \in Po(\eta_l)$ *be independent random variables with poisson distributions for* $l = 1, 2, ...$ *Then the random variables* $X_l(n, d_v, d_c)$ *converge in distribution to* $X_{l\infty}$ *jointly for all* l.

Using Theorem 9 we can compute the probability that a multigraph from ensemble

$X_l(n, d_v, d_c)$ is a simple graph. We can also find the asymptotic distribution of the girth.

Corollary 4. *A multigraph from the ensemble $X_l(n, d_v, d_c)$ is simple with probability converging to*

$$P_{simple} = e^{-\frac{(d_v-1)(d_c-1)}{2}}. \tag{64}$$

Corollary 5. *Let $\eta_l = \frac{\left[(d_v-1)(d_c-1)\right]^l}{2l}$. A multigraph from the ensemble $g_l(n, d_v, d_c)$ has girth greater than or equal to $2g$ with probability converging to*

$$P_{2g} = e^{-\sum_{l=1}^{g} \eta_l}. \tag{65}$$

We now find the distribution of small stopping sets in $g(n, \lambda, \rho)$. First, we prove a lemma.

Lemma 9. *For any constant integer l, with probability $1 - O(\frac{1}{n})$, the multigraph $g(n, \lambda, \rho)$ does not have any stopping set of size l that contains a variable node of degree higher than two.*

Now we can find the probability distribution of small stopping sets in $g(n, \lambda, \rho)$. Let $Y_l(n, \lambda, \rho)$ be the number of stopping sets of size l in $g(n, \lambda, \rho)$. Then, we have [96].

Theorem 10. *Consider the ensemble $g(n, \lambda, \rho)$ with $\rho(x) = x^{d_c-1}$. Let $\mu_l = \frac{\left[\lambda'(0)\rho'(1)\right]^l}{2l}$ and let $Y_{l\infty} \in Po(\mu_l)$ be independent random variables with poisson distributions for $l = 1, 2, ...$ Then the random variables $Y_l(n, \lambda, \rho)$ converge in distribution to $Y_{l\infty}$ jointly for all l.*

Proof. (Sketch) Let $Y_{l,n} = Y_l(n, \lambda, \rho)$. We use the method of moments to prove the theorem. In particular, it suffices to prove

$$E[(Y_{l_1,n})_{s_1}(Y_{l_2,n})_{s_2}...(Y_{l_r,n})_{s_r}] \to \mu_{l_1}^{s_1}\mu_{l_2}^{s_2}...\mu_{l_r}^{s_r} \quad \text{as} \quad n \to \infty. \tag{66}$$

This can be proved using direct computation of $E[(Y_{l_1,n})_{s_1}(Y_{l_2,n})_{s_2}...(Y_{l_r,n})_{s_r}]$ and evaluating it for large n. For example, the simplest case is to show $E[(Y_{l,n})] = \mu_l$. To prove this note that

$$E[(Y_{l,n})] = \binom{\psi_2 n}{l}\binom{m}{l}\frac{l!(l-1)!}{2}\left\{\frac{[2d_c(d_c-1)]^l}{E \times (E-1)...(E-2l+1)}\right\} \tag{67}$$

where $m = (1-R)n$ is the number of check nodes and $E = md_c$ is the number of edges in the graph and ψ_2 is the fraction of variable nodes of degree two. For a constant number l we have

$$\lim_{n\to\infty} E[(Y_{l,n})] = \frac{\left[\frac{2\psi_2(d_c-1)}{d_c(1-R)}\right]^l}{2l} = \frac{\left[\lambda'(0)\rho'(1)\right]^l}{2l}. \tag{68}$$

We can prove the general case (i.e, Equation (67)) similarly; however, more computation is necessary. □

Similar to Corollary 5, we can find the probability distribution of the minimum stopping set in the ensemble $g(n,\lambda,\rho)$.

Corollary 6. *Let* $\mu_l = \frac{\left[\lambda'(0)\rho'(1)\right]^l}{2l}$. *The size of minimum stopping set of a multigraph from the ensemble* $g(n,\lambda,\rho)$ *is greater than or equal to* s *with probability converging to*

$$P_s = e^{-\sum_{l=1}^{s}\mu_l}. \tag{69}$$

It is worth noting that the above corollaries show contiguity of $g(n,\lambda,\rho)$ and certain expurgated ensembles. That is any property that holds asymptotically almost surely for $g(n,\lambda,\rho)$ holds for the others. Thus, for example by Corollary 4, if we know that a randomly chosen multigraph from the ensemble $g(n,\lambda,\rho)$ can be used to achieve arbitrarily small error rate over a symmetric channel with high probability, then the corresponding simple graph ensemble can also achieve arbitrarily small error probability.

Finally, we can conclude the following result that has also been previously given by [109].

Corollary 7. *Consider the ensemble* $g(n,\lambda,\rho)$ *with* $\rho(x) = x^{d_c-1}$ *and* $\lambda'(0)\rho'(1) > 1$. *Let* $\omega(n)$ *tend to infinity arbitrarily slowly, as* n *grows. Then the minimum distance and the minimum stopping set size are less than* $\omega(n)$ *with high probability.*

4.3 Error Floor Due to Small Stopping Sets

In this section we find the average error probability due to small stopping sets in the graphs for memoryless binary-input output-symmetric (MBIOS) channels. This error probability is inversely proportional to n, the code length. Thus, it causes error floor in practical systems

such as page-oriented memories in which the code lengths may be around $n = 10^4$. We note that for ensembles with $\lambda_2\rho'(1) < 1$, we can use the expurgated ensemble and avoid small stopping sets and thus we can avoid the error floor. However, in capacity-achieving ensembles we should have $\lambda_2\rho'(1) > 1$ and thus the error floor is present [109], [120], [83]. For simplicity we first derive the error probability for the binary erasure channel (BEC) and then we generalize it for general MBIOS channels. We note that the error probability analysis in here is different from those found using density evolution. In fact, in density evolution, the effects of small problematic subgraphs is ignored and we are only concerned with the question of whether the average error rate tends to zero or not. However, here our assumption is that we are using the codes below their threshold and we are interested in the error floor effect caused by small problematic subgraphs.

As it was mentioned in the previous sections, small stopping sets in $g(n, \lambda, \rho)$ correspond to cycles consisting of variable nodes of degree two. Since these stopping sets are necessarily disjoint the error probability of their union is equal to the sum of error probabilities caused by each of them. Let ϵ be the erasure probability of the channel. A stopping set of size l with probability ϵ^l causes the error probability l/n. Thus if P_l be the average erasure probability of stopping sets of size l in the graph, we have

$$P_l = \sum_{k=0}^{\infty} \Pr\{Y_l = k\} \frac{\epsilon^l k l}{n} = \frac{l\epsilon^l \mu_l}{n}. \tag{70}$$

Then, the average erasure probability of small stopping sets is given by

$$P_s = \sum_{l=1}^{\infty} P_l = \frac{1}{2n} \sum_{l=1}^{\infty} \left[\epsilon\lambda'(0)\rho'(1)\right]^l. \tag{71}$$

Note that again we used the fact that distinct stopping sets are disjoint (i.e., they do not share any variable nodes). By the stability condition [54] $\epsilon\lambda'(0)\rho'(1) < 1$, thus

$$P_s = \frac{1}{2n}\left(\frac{\epsilon\lambda'(0)\rho'(1)}{1 - \epsilon\lambda'(0)\rho'(1)}\right). \tag{72}$$

Note that for small values of ϵ, P_s is almost proportional to ϵ. This means that P_S decreases slowly as ϵ decreases, and thus it results in an error floor effect. Figure 17 shows the average number of erasures after the iterative decoding for the BEC.

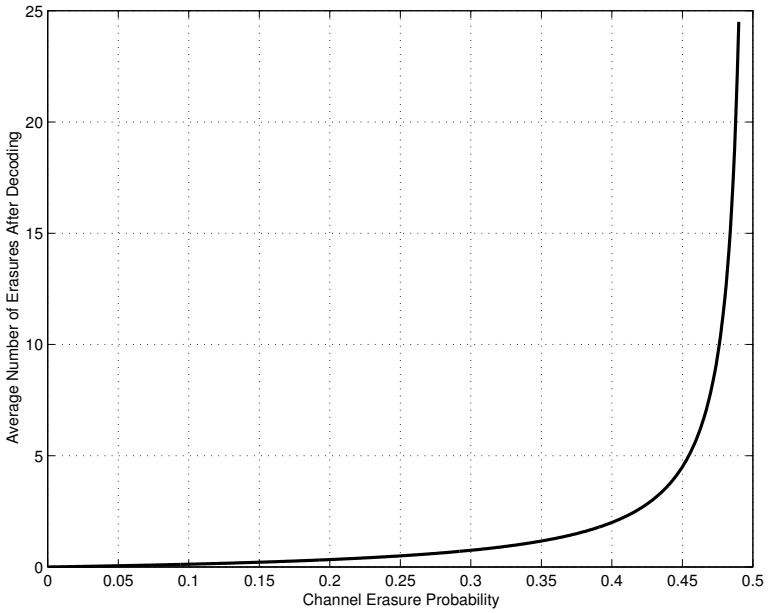

Figure 17: Average number of incorrectly decoded bits for LDPC ensembles with $\lambda'(0)\rho'(1) = 2$ for the BEC.

The above argument can be generalized for other MBIOS channels. Consider a MBIOS channel with parameter θ, where $\theta \in [\theta_{min}, \theta_{max}]$ and $\theta_{min}, \theta_{max} \in \mathbb{R} \cup \{-\infty, +\infty\}$. For example, for the binary-input additive white Gaussian noise (BIAWGN) channel, θ can be considered as the variance σ of the noise. Let $\mathcal{C}$ be a class of channels with parameter θ. Thus, any channel C_θ in $\mathcal{C}$ is uniquely determined by its variable θ. A channel in $\mathcal{C}$ with parameter θ_0 is called C_{θ_0}. The capacity of the channel C_{θ_0} is denoted by c_{θ_0}. For simplicity, we assume that c_θ is a continuous function of θ. Similar to [112], we consider physically degraded channels. For clarity of exposition we assume that if $\theta_1 < \theta_2$, then C_{θ_2} is physically degraded with respect to C_{θ_1}. For the channel C_{θ_0} assuming the all-one code word has been sent, we define the random variable Z_{θ_0} as

$$Z_{\theta_0} = \ln \frac{p(X = 1|Y = y, \theta = \theta_0)}{p(X = -1|Y = y, \theta = \theta_0)}, \tag{73}$$

where X and Y are the input and output of the channel, respectively. Let $Z_\theta^{(l)}$ be a random variable with the same distribution as the sum of l i.i.d random variables each having the same distribution as Z_θ. For example for BIAWGN channel with noise variance σ^2, $Z_\theta^{(l)}$ is a Gaussian random variable with mean l and variance $l\sigma^2$. We have the following lemma [96].

Lemma 10. *Let $v_1, v_2, ..., v_l$ be the variable nodes of degree two in a stopping set S where l, the size of the stopping set, is a positive integer. Then the probability that after the iterative decoding, all the bits corresponding to $v_1, v_2, ..., v_l$ are decoded incorrectly is greater than or equal to $P_e(Z_\theta^{(l)})$.*

Using Lemma 10, we can lower-bound the average probability of small stopping sets as

$$P_s = \sum_{l=1}^{\infty} P_l = \frac{1}{2n} \sum_{l=1}^{\infty} \left[\lambda'(0)\rho'(1)\right]^l P_e(Z_\theta^{(l)}). \tag{74}$$

Again it is easy to show that the above sum is finite using the stability condition. For example for BIAWGN channel with noise variance σ^2, we have

$$P_e(Z_\theta^{(l)}) = \frac{1}{\sqrt{2\pi l\sigma^2}} \int_{-\infty}^{0} \exp\left\{ - \frac{(x-l)^2}{2l\sigma^2} \right\} dx \tag{75}$$

Figure 18 shows the average number of incorrectly decoded bits for LDPC ensembles with $\lambda'(0)\rho'(1) = 1.6854$ over the BIAWGN channel.

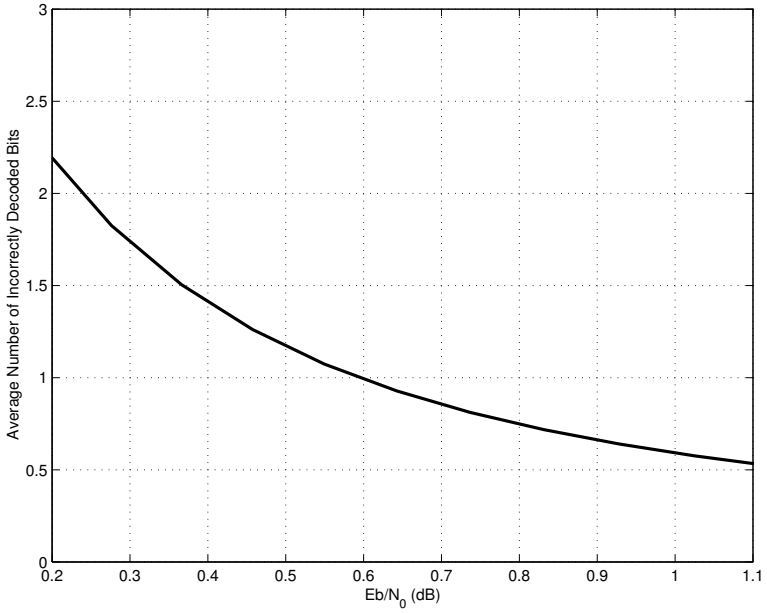

Figure 18: Average number of incorrectly decoded bits for LDPC ensembles with $\lambda'(0)\rho'(1) = 1.6854$ for the BIAWGN channel .

Finally for the binary symmetric channel (BSC) with crossover probability p, we have

$$P_e(Z_\theta^{(l)}) = \sum_{k=\frac{l'+1}{2}}^{l'} \binom{l'}{k} p^k (1-p)^{l'-k} \tag{76}$$

where $l' = l$ if l is odd and $l' = l - 1$ otherwise. Figure 19 shows the average number of incorrectly decoded bits for LDPC ensembles with $\lambda'(0)\rho'(1) = 1.5978$ for the BSC.

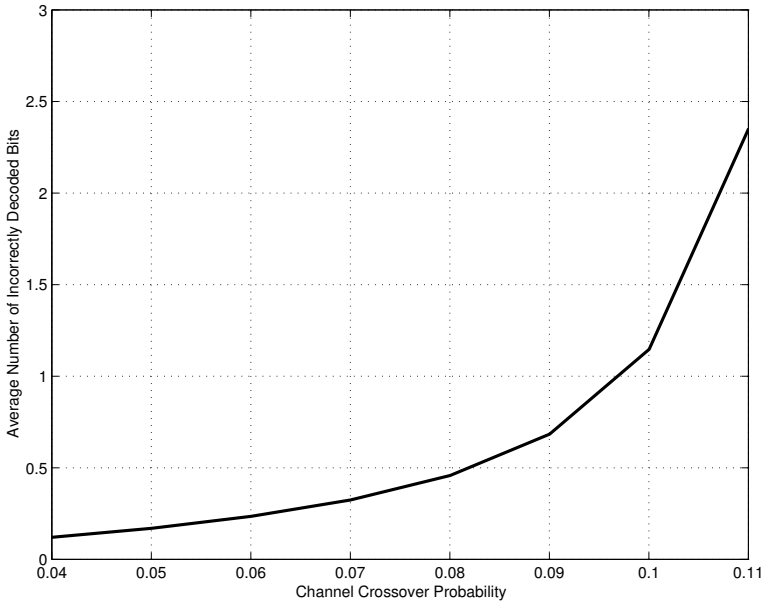

Figure 19: Average number of incorrectly decoded bits for LDPC ensembles with $\lambda'(0)\rho'(1) = 1.5978$ for the BSC.

Let us define the word error probability (WEP) capacity of an ensemble of LDPC codes as the supremum value of the channel parameter such that the word error probability can be made arbitrarily small. It is worth noting that the above arguments shows that the WEP capacity of the conventional ensembles of LDPC codes is bounded away from the Shannon limit. Even expurgation cannot help, because by Corollary 7 the probability of having no small stopping set is not bounded away from zero for the capacity-achieving sequences of code ensembles [120]. However, as we will see in the next section, modified LDPC code ensembles can be capacity-achieving in terms of word error probability. Finally we note that the error floor effect that was studied here is not limited to iterative decoding. In fact, it is an inherent probability of codes and is present in all types of decoding, in particular, maximum likelihood (ML) decoding. The reason is that the small stopping sets that are responsible for the error floor problem, correspond to codewords (Lemma 9).

4.4 Ensembles with Good Error Floor Performance

In this section we discuss two methods to avoid the error floor problem that was studied in the previous section. Our aim is to avoid error floor efficiently. That is, we want to minimize the price we may have to pay for having codes with no error floor. In particular, we want to minimize the possible rate loss or possible increase in complexity. As we mentioned these are key factors in designing codes for page-oriented memories. An obvious solution that comes to mind is to use an outer code with a very high rate. In order to minimize the complexity we choose this outer code to be an LDPC code so that we can combine the two codes to obtain one LDPC code. That is the rows of the parity check matrix of the new code will be the union of the rows of the two codes. We call the new ensemble the modified ensemble. This is very similar to the idea of pre-coding used in raptor codes [117]. Asymptotically the rate of the outer code can be arbitrarily small; thus using this method we can construct capacity-achieving codes with good error floor performance (at least for BEC). However, in practical systems there will be a loss in the rate. Let C_1 and C_2 (the outer code) be the two codes that are combined to make the code C. The code C_2 should be chosen from an ensemble of LDPC codes with linear typical minimum distance. For example one choice is to use the regular $(3, d_c)$ ensemble.

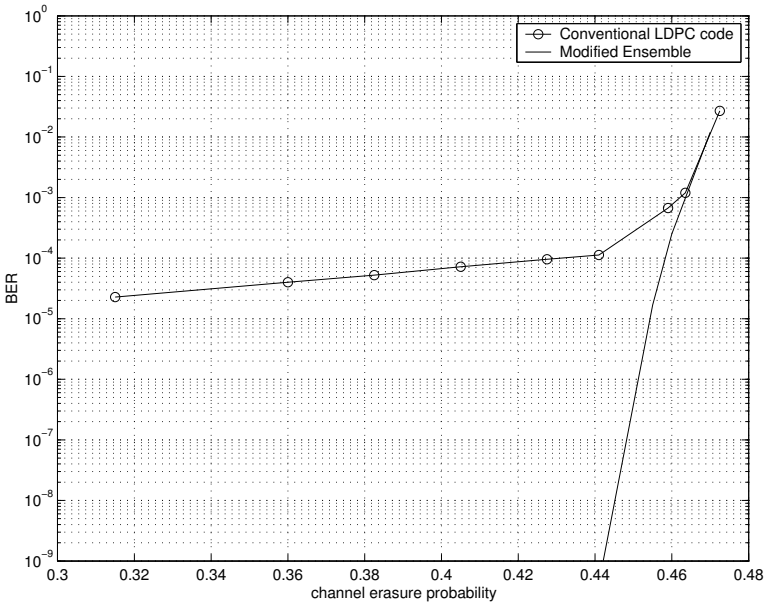

Figure 20: BER of conventional and modified ensembles of LDPC codes.

Figure 20 shows the simulation results for the performance of modified ensemble. In the figure, we show the performance of the code C_1 of length 10000 which is chosen from the ensemble given by the following degree distribution [1]

$$\lambda_1(x) = 0.2498x + .2472x^2 + .1480x^5 + .0033x^6 + .3517x^{19},$$
$$\rho_1(x) = x^7$$

In the figure, it is also shown the performance of a modified code which is obtained by appending a $(3, 120)$-regular code to the code C_1. This results in 5 percent rate loss. However, as we see the error floor performance is improved significantly. It is obvious that the choice of C_2 here is not optimal and can be improved. It is worth noting that the parity-check nodes of the code C_2 do not have to cover all the variable nodes. In fact, by Lemma 9, it suffices to choose C_2 a code whose variable nodes are the degree-two variable nodes of C_1. The above discussion about the lack of error floor in the modified ensemble can be partially made rigorous at least for the BEC as follows [96].

Theorem 11. *For any $R \in (0, 1)$, there exists a sequence $\{C_{1n}, C_{2n}\}_{n=1}^{\infty}$ of the modified ensembles of LDPC codes defined above such that the threshold for word error probability δ_n^{WER} over the erasure channel, satisfies*

$$\lim_{n\to\infty} \delta_n^{WER} = 1 - R. \tag{77}$$

Note that unlike the usual capacity-achieving sequences for which the word error rate is bounded away from zero, Theorem 11 guarantees the existence of capacity-achieving sequences for which the word error rate tends to zero. Although, technically speaking even this result does not guarantee that the codes will not show any error floor effect; however, having zero word error rate certainly implies that the error-floor performance of the codes should be much better.

Another method to avoid error floor is to use codes with linear typical minimum distance. In particular, we may use the suitably expurgated ensembles satisfying $\lambda'(0)\rho'(1) < 1$. However, the flatness theorem [120] implies that these ensembles are not capacity-achieving

at least for the BEC. In the next sections our aim is to study the achievable performance of LDPC codes with linear minimum distance.

4.5 LDPC Codes with Linear Minimum Distance

4.5.1 Lower Bounds on the Achievable Rates

In this section we provide lower bounds on the achievable rates of LDPC codes with linear minimum distance over MBIOS channels [99, 102]. Remember our terminology for MBIOS channels in Section 4.3. We consider MBIOS channels with parameter θ, where $\theta \in [\theta_{min}, \theta_{max}]$ and $\theta_{min}, \theta_{max} \in \mathbb{R} \cup \{-\infty, +\infty\}$. Also, $\mathcal{C}$ shows is a class of channels with parameter θ. Thus, any channel C_θ in $\mathcal{C}$ is uniquely determined by its variable θ. A channel in $\mathcal{C}$ with parameter θ_0 is called C_{θ_0}. The capacity of the channel C_{θ_0} is denoted by c_{θ_0}. We assume that if $\theta_1 < \theta_2$, then C_{θ_2} is physically degraded with respect to C_{θ_1}. Let $F_0(x;\theta)$ be the distribution function of Z_θ (defined in Section 4.3) under the assumption that a "1" is transmitted. Similar to [54], we define

$$r(\theta) = -\ln\left(\int_{\mathbb{R}} e^{-\frac{x}{2}} d(F_0(x;\theta))\right). \tag{78}$$

For any $\theta \in [\theta_{min}, \theta_{max}]$, let α_θ be the supremum value of R/c_θ for which there exists an ensemble (λ, ρ) of LDPC codes that has rate R and threshold (under belief propagation decoding) higher than or equal to θ. For the BEC we have $\alpha_\theta = 1$ where θ is the erasure probability [70], [120], [118], [84]. For other MBIOS channels we know $0 \leq \alpha_\theta \leq 1$ and it is conjectured that $\alpha_\theta = 1$ for all θ. Let R_θ be the supremum value of R, the rate of an ensemble (λ, ρ) of LDPC codes with the threshold higher than or equal to θ satisfying $\lambda'(0)\rho'(1) < 1$. Shokrollahi's flatness theorem [120] implies that $R_\theta < c_\theta$ for the BEC. Thus, we are sure that unlike the general class of LDPC codes, the LDPC codes with typical linear minimum distance are not capacity achieving. Moreover, it is conjectured in [120] that the stability condition is satisfied with equality for capacity-achieving LDPC codes over other MBIOS channels. If this is the case, then $R_\theta < c_\theta$ for all MBIOS channels. However, one of the results of this chapter is that we do not lose too much by restricting to the codes with the linear minimum distance constraint. We first prove two lemmas.

Lemma 11. *Let (λ, ρ) be an ensemble of LDPC codes having the threshold θ_{th} under belief propagation decoding. For $0 \leq \tau < \lambda_2$ define λ^τ and ρ^τ as follows*

$$\begin{aligned} \lambda^\tau(x) &= (\lambda_2 - \tau)x + (\lambda_3 + \tau)x^2 + \sum_{i>3} \lambda_i x^{i-1}, \\ \rho^\tau(x) &= \rho(x) = \sum_i \rho_i x^{i-1}. \end{aligned} \tag{79}$$

Then the threshold of the ensemble $(\lambda^\tau, \rho^\tau)$ is greater than or equal to θ_{th}.

Proof. Let $\theta < \theta_{th}$ and P_0 be the density function of Z_θ. Then the density evolution formulas for the ensemble (λ, ρ) can be described as [54]

$$P_l = P_0 \bigotimes \lambda(Q_l) \tag{80}$$

$$Q_l = \Gamma^{-1}(\rho(\Gamma(P_{l-1}))). \tag{81}$$

For the ensemble $(\lambda^\tau, \rho^\tau)$, we introduce a message-passing decoding algorithm named Algorithm B. In this algorithm, for any edge e that connects a variable node v and a check node c, the messages are computed in the following way in each iteration. The message from c to v is computed the same way as the standard belief propagation algorithm. If the degree of v is not equal to 3, then the message from v to c is also computed the same way as the standard belief propagation algorithm. However, if the degree of v is equal to 3 then there are two other edges (e_1 and e_2) incident with v. In this case, with probability $\frac{\tau}{\lambda_3^\tau} = \frac{\tau}{\lambda_3 + \tau}$, we choose one of the edges e_1 and e_2 at random. Suppose we choose e_1. Then, we compute the message from v to c similar to the belief propagation algorithm except that we disregard e_1 in the computation. In other words, the message from v to c is computed based on the observation from the channel and the message transmitted to v by e_2 in the previous iteration. Now if we obtain the density evolution formulas for Algorithm B on the ensemble $(\lambda^\tau, \rho^\tau)$, we get the same equations as (128) and (130). This shows that when $\theta < \theta_{th}$, the error probability of Algorithm B on the ensemble $(\lambda^\tau, \rho^\tau)$ tends to zero as the number of iteration goes to infinity. But, based on the cycle-free-neighborhood lemma in [112], the belief propagation algorithm has an asymptotic error rate less than or equal to Algorithm

B. Thus we conclude when $\theta < \theta_{th}$, the error probability of the belief propagation decoding on the ensemble $(\lambda^\tau, \rho^\tau)$ tends to zero as the number of iteration goes to infinity. Therefore, the threshold of the ensemble $(\lambda^\tau, \rho^\tau)$ under belief propagation decoding is greater than or equal to θ_{th}. □

The intuition behind Lemma 11 is that we change degree-two variable nodes into degree-three variable nodes without changing the check node degree distribution. We note that the lemma states that the threshold can only improve. The reason for this improvement is that the rate of the code is getting worse, as more bits receive more information.

Lemma 12. *For any ensemble (λ, ρ) of LDPC codes, we have*

$$\frac{1}{2\rho'(1)} < \int_0^1 \rho(x)dx. \tag{82}$$

Proof. The function $\rho(x)$ has the following properties:

$$\rho(0) = 0, \quad \rho(1) = 1, \quad \rho^{(n)}(x) > 0 \ \text{ for } \ x \in (0, 1] \text{ and } \ 0 \leq n \leq d_{c_{max}} \tag{83}$$

where $\rho^{(n)}(x)$ is the n'th derivative of the function ρ and $d_{c_{max}}$ is the largest degree of check nodes. Figure 67 shows the plot of a typical $\rho(x)$. The tangent line to the curve at point $(1, 1)$ is also shown in the figure. We have

$$1 - x_0 = \frac{1}{\rho'(1)}. \tag{84}$$

Therefore, the area of the triangle ABC in Figure 67 is equal to $\frac{1}{2\rho'(1)}$. Since $\rho(x)$ is a convex function in $[0, 1]$, the area of the triangle ABC is less than the area under the curve. That is, we have

$$\frac{1}{2\rho'(1)} < \int_0^1 \rho(x)dx. \tag{85}$$

□

Now, we can state and prove the following theorem that gives a lower bound on the achievable rates of LDPC codes over general MBIOS channels [99, 102].

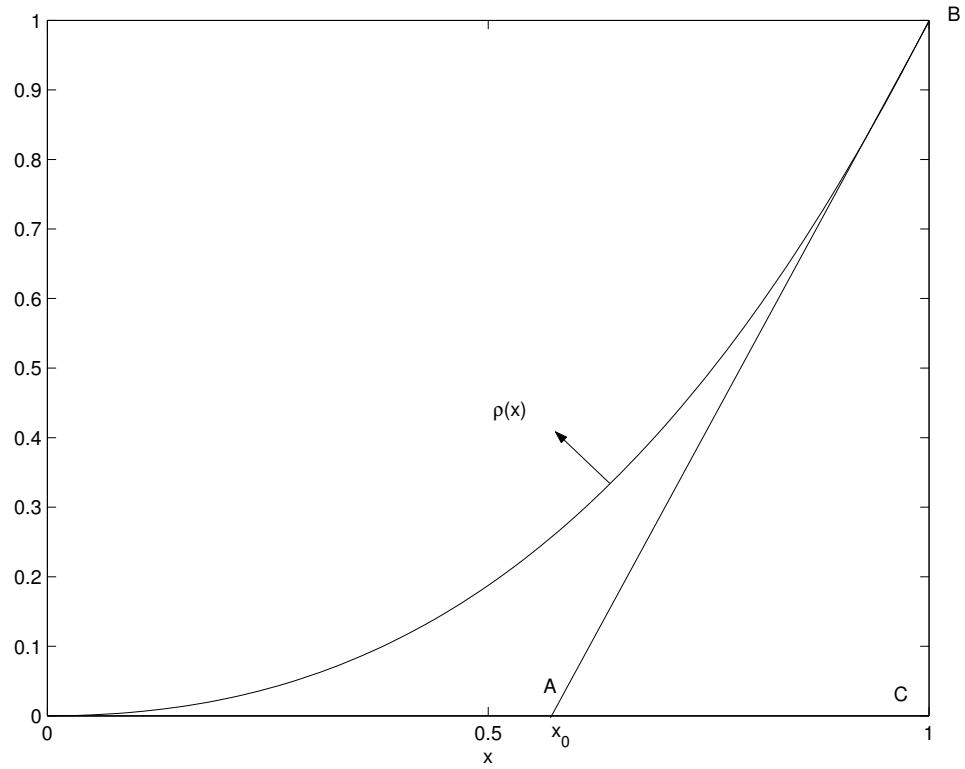

Figure 21: Plot of the function $\rho(x)$.

Theorem 12. *For any MBIOS channel with parameter θ and capacity c_θ, we have*

$$R_\theta \geq 1 - \frac{1 - \alpha_\theta c_\theta}{1 - \frac{(1-\alpha_\theta c_\theta)(e^{r(\theta)}-1)}{3}}. \tag{86}$$

Proof. Let (λ, ρ) be an ensemble of LDPC codes with $\lambda'(0)\rho'(1) \geq 1$ having the threshold θ_{th}. Similar to Lemma 11, define λ^τ and ρ^τ as follows

$$\begin{aligned}
\lambda^\tau(x) &= (\lambda_2 - \tau)x + (\lambda_3 + \tau)x^2 + \sum_{i>3} \lambda_i x^{i-1}, \\
\rho^\tau(x) &= \rho(x) = \sum_i \rho_i x^{i-1}, \qquad (87) \\
\tau &> \frac{\lambda'(0)\rho'(1) - 1}{\rho'(1)}. \qquad (88)
\end{aligned}$$

Then by Lemma 11, the ensemble $(\lambda^\tau, \rho^\tau)$ has a threshold greater than or equal to θ_{th}. It also satisfies $\lambda'^{\tau}(0)\rho'^{\tau}(1) < 1$. Now we claim that for any $\xi > 0$, by a suitable choice of τ,

the ensemble has the rate R^τ satisfying

$$R^\tau \geq 1 - \frac{1-R}{1-\frac{(1-R)(e^{r(\theta)}-1)}{3}} - \xi. \tag{89}$$

Assuming this claim, and because for any θ there exists an ensemble (λ, ρ) of LDPC codes with the rate arbitrary close to $\alpha_\theta c_\theta$, we conclude the theorem. It should be noted that if the ensemble that achieves the rate $\alpha_\theta c_\theta$ satisfies $\lambda'(0)\rho'(1) < 1$, then the assertion of the theorem is trivial. Thus we may assume without loss of generality that $\lambda'(0)\rho'(1) \geq 1$. It remains to prove the claim. The rate R^τ can be expressed as

$$R^\tau = 1 - \frac{\int \rho^\tau}{\int \lambda^\tau} \tag{90}$$

$$= 1 - \frac{(1-R)\int \lambda}{\int \lambda - \frac{\tau}{6}}, \tag{91}$$

where the integrals are on $[0, 1]$. Using the stability condition [54] and Lemma 12, we have

$$\frac{\lambda'(0)\rho'(1)-1}{\rho'(1)} \leq \frac{(e^{r(\theta)}-1)}{\rho'(1)} \leq 2(e^{r(\theta)}-1)\int \rho = 2(1-R)(e^{r(\theta)}-1)\int \lambda. \tag{92}$$

Thus by choosing τ close enough to $\frac{\lambda'(0)\rho'(1)-1}{\rho'(1)}$, we can ensure that

$$R^\tau \geq 1 - \frac{1-R}{1-\frac{(1-R)(e^{r(\theta)}-1)}{3}} - \xi. \tag{93}$$

□

It is worth noting that Theorem 12 not only gives a lower bound on the achievable rate, but also gives a distribution meeting the lower bound. However, we can find this distribution only if we know codes that approach the optimal rate $\alpha_\theta c_\theta$. We also note that, the basic idea behind Theorem 12 is to start with an optimized degree distribution without any constraint on $\lambda'(0)\rho'(1)$. Then using Lemma 11 we transform the degree distribution into one with $\lambda'(0)\rho'(1) < 1$. Using this method we can find an analytical lower bound on the achievable rate. However, in practice, one may try optimize the degree distribution while imposing $\lambda'(0)\rho'(1) < 1$ as a constraint.

For the BIAWGN channel we let θ be σ, the variance of noise, and the lower bound becomes

$$R_\sigma \geq 1 - \frac{1 - \alpha_\sigma c_\sigma}{1 - \frac{(1-\alpha_\sigma c_\sigma)(e^{\frac{1}{2\sigma'^2}}-1)}{3}}. \tag{94}$$

Using the lower bound on the rate, we can find an upper bound on the gap between the ensemble threshold and the Shannon limit for the BIAWGN channels. Figure 22 shows this upper bound for the BIAWGN channel assuming $\alpha_\theta = \alpha_\sigma = 1$ for any $\sigma \in [0, \infty)$ (i.e, assuming that LDPC codes are capacity achieving over BIAWGN channels). Therefore, as shown in the figure, by restricting to LDPC codes with the linear minimum distance property, we lose at most $1.1dB$. It is worth noting that in storage systems we usually use high-rate codes [107]. Examining Figure 22 reveals that the gap is very small at these rates. For example, the gap is less than $.4dB$ for the rate $R = .9$. This is important from the practical point of view.

For the binary symmetric channel (BSC) channel, we let θ be the crossover probability p. Thus, using Theorem 12, the lower bound becomes

$$R_p \geq 1 - \frac{1 - \alpha_p c_p}{1 - \frac{(1-\alpha_p c_p)(\frac{1}{2\sqrt{p(1-p)}}-1)}{3}}. \tag{95}$$

Figure 23 shows this lower bound for the BSC assuming $\alpha_\theta = \alpha_p = 1$ for any $p \in [0, 1)$ (i.e, assuming LDPC codes are capacity achieving over the BSC). The figure suggests that the rate loss due to the linear minimum distance property is small.

For the binary erasure channel (BEC), we can find a better bound than the one given by Theorem 12. In the following, we exploit the developed theory on the capacity achieving sequences to obtain a tighter bound [99]. Furthermore, we can explicitly find sequences of LDPC codes meeting this lower bound.

Theorem 13. *For any BEC with erasure probability δ we have*

$$R_\delta \geq \frac{5(1-\delta)}{\delta + 5}. \tag{96}$$

Proof. For any BEC with erasure probability δ, we construct a sequence of LDPC code ensembles with linear typical minimum distance and threshold greater than or equal to δ whose rates approach $\frac{5(1-\delta)}{\delta+5}$. Our construction is based on right-regular LDPC codes.

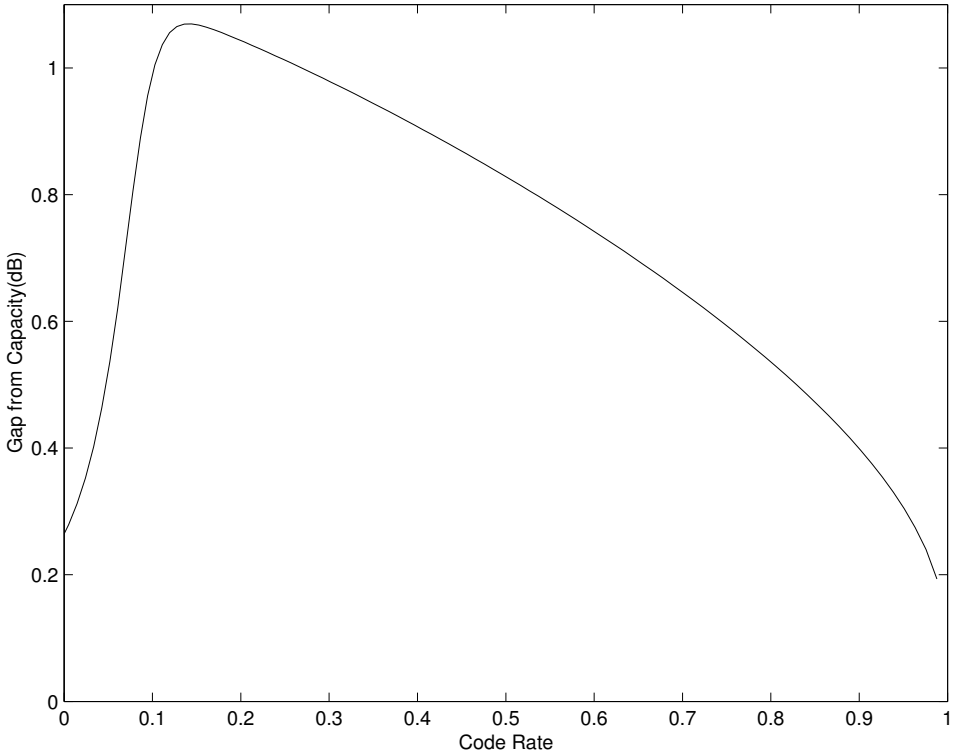

Figure 22: Upper bound on the gap between the ensemble threshold and the Shannon limit of the BIAWGN channel for LDPC code ensembles with linear typical minimum distance. The bound is obtained using the lower bound on the rate given by (94).

Choose $R < 1 - \delta$. As it is shown in [118] and [84], there exists a sequence $\{(\lambda_n, \rho_n)\}_{n=1}^{\infty}$ of right-regular LDPC codes of rate R and thresholds $\delta_{n,th}$ as follows

$$\lambda_n(x) = \sum \lambda_{i,n} x^{i-1}, \tag{97}$$

$$\rho_n(x) = x^{a_n}, \tag{98}$$

$$\lim_{n \longrightarrow \infty} a_n = \infty, \tag{99}$$

$$\lim_{n \longrightarrow \infty} \delta_{n,th} = 1 - R. \tag{100}$$

Using the Flatness Theorem we obtain

$$\lim_{n \longrightarrow \infty} \lambda_{2,n} \rho_n'(1) = \lim_{n \longrightarrow \infty} \lambda_{2,n} a_n = \frac{1}{1-R}. \tag{101}$$

Now we construct the sequence of $\{(\lambda_n^{\tau_n}, \rho_n^{\tau_n})\}_{n=1}^{\infty}$ of LDPC code ensembles with thresholds $\delta_{th}^{\tau_n}$ and rates $R_n^{\tau_n}$ as follows

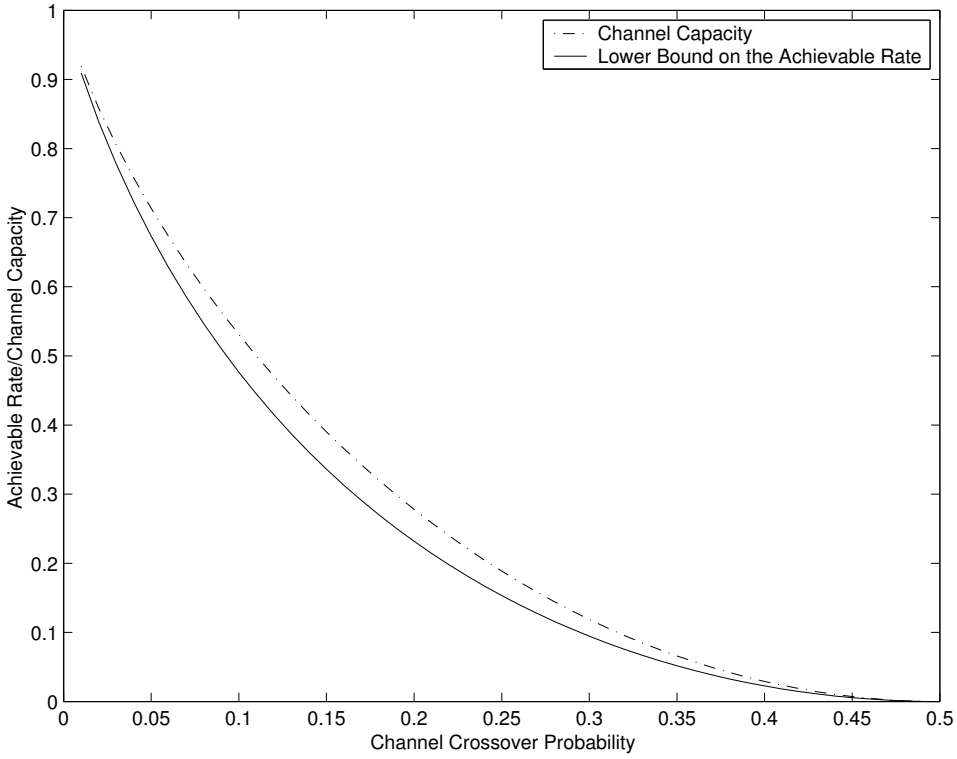

Figure 23: Lower bound on the achievable rate for LDPC codes with linear minimum distance on the BSC.

$$\lambda_n^{\tau_n}(x) = \sum \lambda_{i,n}^{\tau_n} x^{i-1} = (\lambda_{2,n} - \tau_n)x + (\lambda_{3,n} + \tau_n)x^2 + \sum_{i>3} \lambda_{i,n} x^{i-1}, \tag{102}$$

$$\rho_n^{\tau_n}(x) = \rho_n(x) = x^{a_n}, \tag{103}$$

$$\tau_n = \frac{\lambda_{2,n} a_n - 1}{a_n} + \frac{1}{n(a_n+1)}. \tag{104}$$

We first notice that by Lemma 11 we have $\delta_{th}^{\tau_n} \geq \delta_{n,th}$. Since $\lim_{n \longrightarrow \infty} \delta_{n,th} = 1 - R > \delta$, for some $N > 0$ we have

$$n > N \Longrightarrow \delta_{th}^{\tau_n} \geq \delta. \tag{105}$$

We also note that

$$0 \leq \tau_n \leq \lambda_{2,n}, \tag{106}$$

$$\lambda_{2,n}^{\tau_n} \rho_n'^{\tau_n}(1) = 1 - \frac{a_n}{n(a_n+1)} < 1, \tag{107}$$

$$\lim_{n \longrightarrow \infty} \tau_n (a_n + 1) = \frac{1}{1-R} - 1. \tag{108}$$

The rate $R_n^{\tau_n}$ of the ensemble $(\lambda_n^{\tau_n}, \rho_n^{\tau_n})$ is given by

$$R_n^{\tau_n} = 1 - \frac{1-R}{1-\frac{\tau_n}{6}(a_n+1)(1-R)}. \tag{109}$$

Thus we have

$$\lim_{n\longrightarrow\infty} R_n^{\tau_n} = 1 - \frac{1-R}{1-\frac{(\frac{1}{1-R}-1)(1-R)}{6}}. \tag{110}$$

Letting R tend to $1-\delta$, we have $R_n^{\tau_n} \longrightarrow \frac{5(1-\delta)}{\delta+5}$ which concludes the theorem. □

Figure 24 shows this lower bound for the BEC. We note that the lower bound on the achievable rate is very close to the capacity. Since capacity achieving right-regular sequences are known [118] and [84], we can explicitly construct the sequences of LDPC codes satisfying the bound given by Theorem 13. It is also easy to see that applying the above procedure to the Tornado sequence [70] results in the same bound. More generally we have the following corollary.

Corollary 8. *If $\{(\lambda_n, \rho_n)\}_{n=1}^{\infty}$ is a capacity achieving sequence of rate R, and $b = \limsup_{n\longrightarrow\infty} \frac{\lambda_{2,n}}{\int \lambda_n}$, then applying the procedure in the proof of Theorem 13, we can find a sequence of LDPC codes satisfying the linear-minimum-distance property having rates $R_n^{\tau_n}$ such that*

$$R_n^{\tau_n} \longrightarrow 1 - \frac{6\delta}{6-(1-\delta)b}. \tag{111}$$

In the following, we show that we can tighten the lower bound of Theorem 13 for the BEC by including the ensemble of punctured LDPC codes. This can be done by choosing an optimized parent LDPC code that has the linear minimum distance property. Since there is no $\frac{R}{c}$ loss due to puncturing [100], we can obtain higher rate LDPC codes with the linear distance property if the puncturing fraction q is less than P defined in Theorem 14. It can be concluded that the resulting bound on $\frac{R}{c}$ can improve upon Theorem 13.

Let us define the performance of any ensemble $\mathcal{C}$ of LDPC codes over the erasure channel by

$$\eta_{\mathcal{C}} = \frac{R}{C} = \frac{R}{1-\delta_{th}} \tag{112}$$

where δ_{th} is the threshold of the code under the standard iterative decoding over the BEC. For $\nu \in (0,1)$, let b_ν be the asymptotic average distance distribution defined in [66]. Then, we have the following theorem [99].

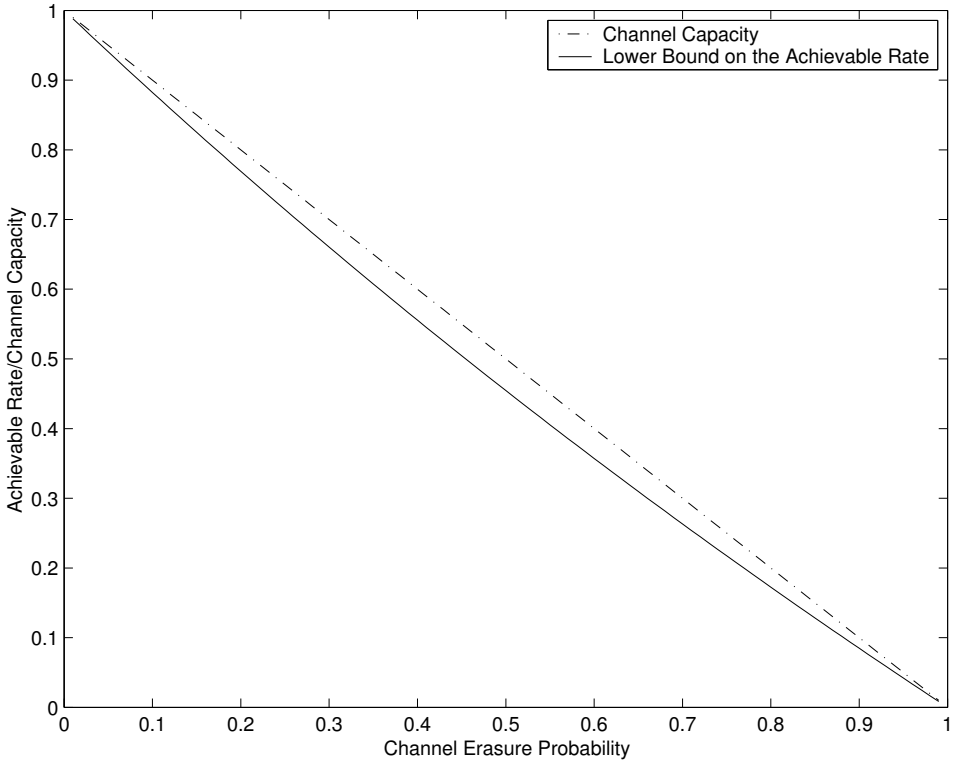

Figure 24: Lower bound on the achievable rate for LDPC codes with linear minimum distance on the BEC.

Theorem 14. *Let (λ, ρ) be an ensemble of LDPC codes of rate R with typical relative minimum distance ν^* (i.e., $d_{min} \geq \nu^* n$ with high probability for the expurgated ensemble, where n is the block length) and $\frac{R}{C} = \frac{R}{1-\delta_{th}} = \eta_0$. Let*

$$P = \sup\left\{p: \ b_\nu + \nu \ln(p) < 0, \ \forall \nu \in [\nu^*, 1]\right\}. \tag{113}$$

Then for rates r satisfying $R \leq r < \frac{R}{1-P}$, there exist punctured LDPC codes whose performance over the erasure channel satisfies $\eta \geq \eta_0$ and has linear typical minimum distance.

Proof. Let $q < P = \sup\left\{p: \ b_\nu + \nu \ln(p) < 0, \ \forall \nu \in [\nu^*, 1]\right\}$. We perform the following experiment. We choose a code from the expurgated ensemble (λ, ρ) at random. Let h_i be the i'th column of the corresponding $m \times n$ parity-check matrix H. We then puncture each bit in the codeword independently with probability q [46]. Puncturing a bit in the codeword can be viewed as erasing the corresponding column of the matrix. Let $A \subset \{1, 2, ..., n\}$ with $|A| = l = \gamma n$, $0 < \gamma < 1$, and E_A be the event that $\sum_{i \in A} h_i = 0$. Let also Q_A be the number

of erased columns in the set A. Then for $0 \leq \zeta < \gamma$, we have

$$\begin{aligned}\Pr\{Q_A \geq (\gamma-\zeta)n\} &= \sum_{i=(\gamma-\zeta)n}^{l} \binom{l}{i} q^i (1-q)^{l-i} \\ &\leq \frac{1}{\sqrt{2\pi}} \frac{1-\frac{\zeta}{\gamma}}{(1-\frac{\zeta}{\gamma}-q)\sqrt{n(\gamma-\zeta)\frac{\zeta}{\gamma}}} \left[\frac{1-\frac{\zeta}{\gamma}}{q}\right]^{-(1-\frac{\zeta}{\gamma})\gamma n} \left[\frac{(1-q)\gamma}{\zeta}\right]^{\zeta n} \quad (114)\end{aligned}$$

where we used Theorem 1.1 of [10]. If p_d is the probability that the minimum distance d is sublinear, then p_d is upper bounded by

$$o(1) + \sum_{A \subset \{1,2,..,n\}, |A| = \gamma n \geq \nu^*} \Pr\{E_A\} \Pr\{Q_A \geq (\gamma-\zeta)n\}. \quad (115)$$

Let $c_s = \sup\left\{ b_\nu + \nu \ln(q), \nu \in [\nu^*, 1] \right\}$. Since $q < P$, we have $c_s < 0$. Applying (114), and letting ζ tend to zero we obtain

$$p_d = o(1) + O(e^{\frac{nc_s}{2}}) \to 0 \text{ as } n \to \infty. \quad (116)$$

□

Using the $(3, 6)$ regular ensemble as an example, we find that for the rates $.5 \leq r < .8469$, there exist punctured LDPC codes with $\eta \geq .8763$ and linear typical minimum distance. By finding good degree distributions we can tighten the lower bound on the achievable rates using Theorem 14.

As we mentioned, for ordinary LDPC code ensembles, the conditions for linear typical minimum distance and linear typical minimum stopping set size are the same. However, for a punctured ensemble this may not be the case. Thus, it would be desirable to obtain similar results to Theorem 14 for the codes with linear minimum stopping set size, rather than the linear minimum distance. In fact, this can be done by replacing the b_v function with the stopping set distribution of the ensemble found in [83].

4.5.2 Upper Bounds on the Achievable Rates

In this section we provide upper bounds on the achievable rates using LDPC codes with linear typical minimum distance over the BEC [99,102]. In [8], authors derived upper bounds on the achievable rates of LDPC codes over the BEC given their right-degree distribution.

We derive similar bounds for LDPC codes with linear minimum distance. By comparing our bounds with the bounds in [8], we get an estimate of the rate loss due to the linear minimum distance constraint. As in [8], it suffices to consider only the case $\delta\rho'(1) > 1$, where δ is the channel erasure probability.

Theorem 15. *For any ensemble (λ, ρ) of LDPC codes with fixed $\lambda'(0) = \lambda_2$ and $\rho'(1)$ having a threshold over the BEC higher than or equal to δ we have*

$$R \leq 1 - \frac{\delta}{1-(1-\delta)^{2\rho'(1)}}\left[1 + \frac{(1-\lambda_2\rho'(1)\delta)^3}{3\delta^3\rho'(1)^3(1-\lambda_2)^2}\right]. \tag{117}$$

Proof. For a given right-degree distribution function $\rho(x)$ we define $y_\rho(x)$ as

$$y_\rho(x) = \frac{1-\rho^{-1}(1-x)}{\delta}. \tag{118}$$

As it is shown in [8], we have

$$\frac{1}{\delta} - \frac{1}{1-R} = \frac{1}{\int \rho}\int (y-\lambda) \tag{119}$$

where the integrals are taken over the interval $[0,1]$. Let $t(x)$ be the tangent line to $y(x)$ at the origin. Moreover, we define

$$\alpha(x) = \lambda_2 x + (1-\lambda_2)x^2. \tag{120}$$

Then we have $\alpha(x) \geq \lambda(x)$ for $x \in [0,1]$. Computing the shaded area in Figure 25 and applying Lemma 12, we obtain the bound in the theorem. □

Now if we consider the ensemble (λ, ρ) of LDPC codes having the linear minimum distance property, we would have $\rho'(1) < \frac{1}{\lambda_2}$. Thus, we have the following corollary.

Corollary 9. *For any ensemble of LDPC codes with $\lambda'(0) = \lambda_2$, a linear typical minimum distance, and a threshold over the BEC higher than or equal to δ, we have*

$$R \leq 1 - \frac{\delta}{1-(1-\delta)^{\frac{2}{\lambda_2}}}\left[1 + \frac{(1-\delta)^3\lambda_2^3}{3\delta^3(1-\lambda_2)^2}\right]. \tag{121}$$

We note that this inequality is similar to the bound given by [118] and the zero-order bound of [8]; however, it has an extra term which is due to the linear-minimum distance property. Now, as in [8], we consider the ensemble of LDPC codes with a fixed given right degree distribution and obtain an upper bound on the achievable rate.

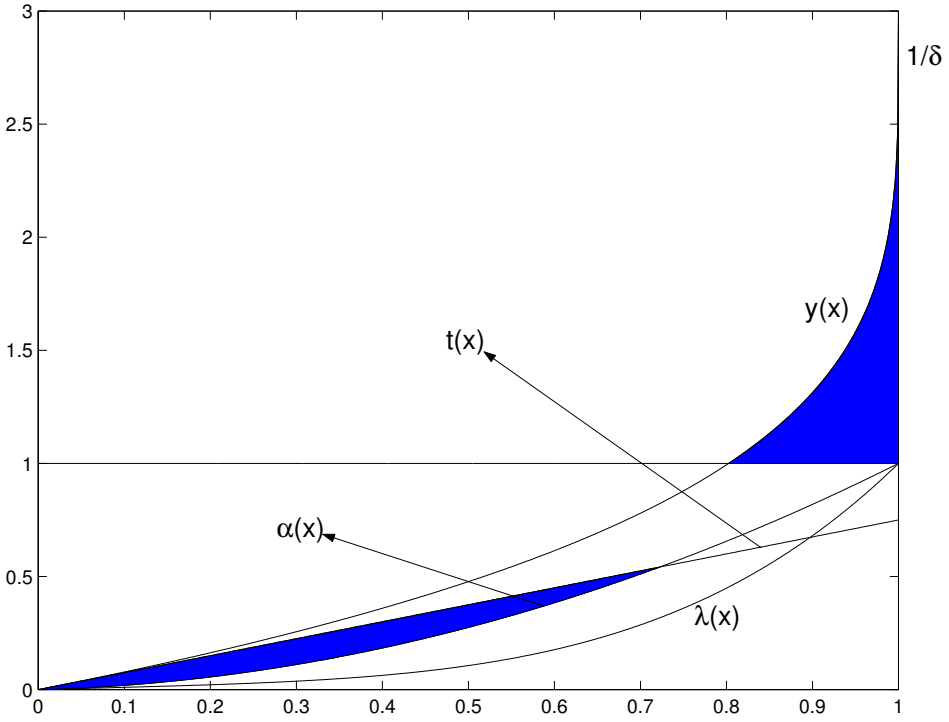

Figure 25: Upper bound on the achievable rate for the LDPC codes with the linear minimum distance property.

Theorem 16. *Consider an ensemble* (λ, ρ) *of LDPC codes with threshold higher than* δ *over the BEC that has a linear typical minimum distance. Define*

$$b(x) = \frac{1}{\rho'(1)} x + (1 - \frac{1}{\rho'(1)}) x^2. \tag{122}$$

Let $c(x) = \min\{y_\rho(x), b(x)\}$ *in* $[0, 1]$. *Let also* $f_\rho(x) = y_\rho(x) - c(x)$. *Then, we have*

$$R \leq 1 - \frac{\delta}{1 - \frac{\delta}{\int \rho} \int f_\rho} \tag{123}$$

where the integrals are taken over the interval $[0, 1]$.

Proof. The condition $\lambda'(0)\rho'(1) < 1$ implies that $b(x) \geq \alpha(x)$. Thus $b(x) \geq \lambda(x)$. Computing the area between $y_p(x)$ and $b(x)$ concludes the theorem. □

Similar to the arguments in the second-order bound of [8], the bound in Theorem 16 can be improved. However, to compare the performance of the code ensembles having the linear typical minimum distance with ones with no restriction, it suffices to work with this simple bound. Figure 26 shows the upper bound of Theorem 16 and the second-order bound of [8]

for right-regular codes over the BEC with erasure probability $\delta = .15$. As it is shown in [8], the second-order bound is tight at least for our example. Thus, the difference between the two curves shows the rate loss because of the linear minimum distance constraint.

At the end, we note that all the results in the chapter can be easily generalized for the condition $\lambda'(0)\rho'(1) \leq a$, where a is a given number. For example, if we want to avoid degree-two variable nodes completely, we should have $\lambda_2 = 0$ (i.e., $a = 0$, in this case there is no need for expurgation and the ensemble has linear minimum distance with high probability). Thus, using bounds similar to the ones in the chapter we can estimate the achievable rates.

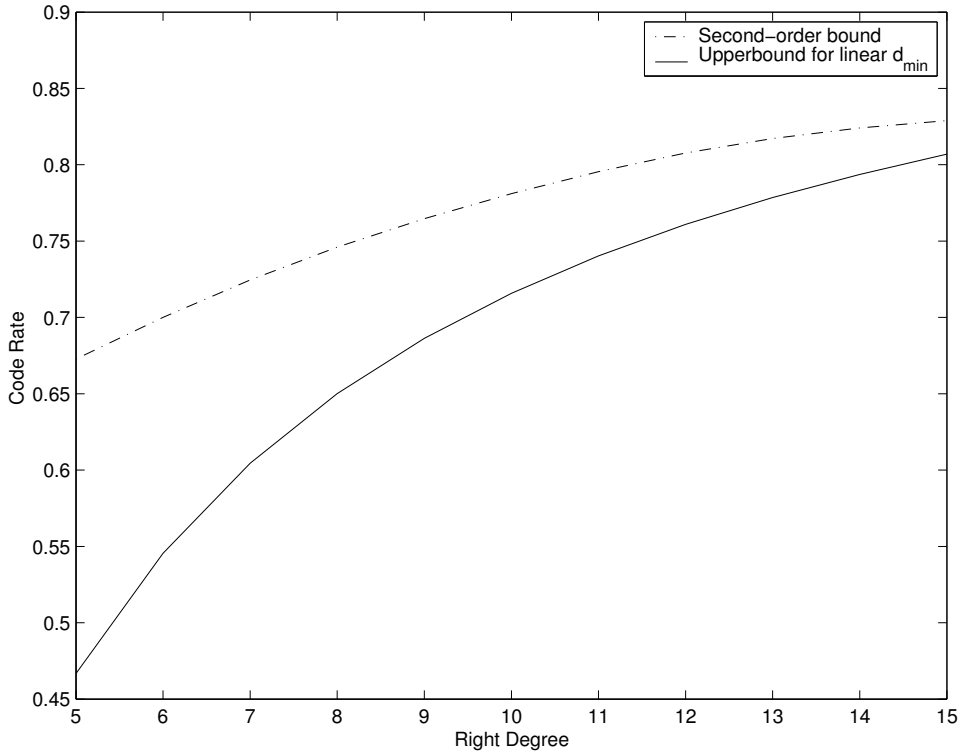

Figure 26: Comparison between the upper bound on the achievable rate for the LDPC codes with the linear minimum distance property and the bound for the unconstrained codes.

4.6 Conclusion

In this chapter we studied some graph theoretic properties of ensembles of LDPC codes. In particular, we derived the asymptotic probability distributions of small cycles and stopping

sets and discussed corollaries of these results. Using, the distribution of small stopping sets we derived lower bounds on error floor probabilities for LDPC codes over general MBIOS channels. We then discussed practical methods to avoid the error floor problem. In particular, we studied performance of LDPC codes that have linear minimum distance.

We derived lower and upper bounds on the achievable rates of the iterative decoding of LDPC code ensembles having linear typical minimum distance and linear typical minimum stopping set. We also gave a design methodology to construct codes meeting the lower bound for the binary erasure channel. We showed that practically the rate of the linear-minimum-distance codes are close enough to the Shannon limit. For example, on the BIAWGN channel, there is at most $1.1dB$ loss due to the linear minimum distance property. Moreover, the loss is much smaller at higher rates. This result implies that it is possible to design codes with low error floors whose rates are close to the capacity. On the other hand, our results on the upper bound for the BEC indicate that if the average right degree is not sufficiently large, the loss can be considerable. This was shown by comparing the upper-bound derived in this chapter with a known tight bound on the rate of LDPC codes.

CHAPTER V

NON-UNIFORM ERROR CORRECTION USING LDPC CODES

5.1 Introduction

In this chapter, we study three closely related applications of low-density parity-check (LDPC) codes: coding for non-uniform channels, rate-compatible coding using punctured codes, and unequal error protection [104–107]. In the first application, we concentrate on the design and analysis of LDPC codes over non-uniform channels. Specifically, we focus on volume holographic memory (VHM) systems that can be modeled as a set of parallel channels as in Fig. 14. We show that using proper LDPC codes instead of conventional coding schemes can result in more than a 50 percent increase in the storage capacity of these systems [104, 107]. In the second application, we investigate punctured LDPC codes and show that they can be considered as a special case of our model for non-uniform channels [106]. Finally, we study unequal error protection using LDPC codes [106].

First, we investigate the design of LDPC codes over a set of parallel subchannels. Consider Fig. 14 where we transmit bits over several binary-input output-symmetric channels. For simplicity, we may assume that the channels are independent. One trivial approach is to design a separate error correcting code for each of the channels. Here, we are interested in designing only one LDPC code as shown in Fig. 14. Suppose we use a code of length n. We transmit any codeword over the set of channels such that $n^{(j)}$ bits in any codeword are transmitted over the jth channel. Let $p_j = \frac{n^{(j)}}{n}$. Assume $0 < p_j$ for $j = 1, ..., k_r$. Let z_j be the random variable that is equal to the log likelihood ratio of a received bit from the jth channel. Then, if the bits that are transmitted over the different channels are chosen randomly from the n bits in a codeword, the set of parallel channels can be modeled as a

single channel having the log likelihood ratio that has the distribution

$$P_Z(z) = \sum_{j=1}^{k_r} p_j P_{Z_j}(z). \tag{124}$$

Therefore, good degree distributions for LDPC codes can be found by the methods described in [112] and [54] for the corresponding $P_Z(z)$.

The first goal of this chapter is to show that for certain practical problems, we can employ an improved method that provides some advantages over the above method. We consider VHM systems and show that they can be modeled as a set of parallel channels as in Fig. 14. Then, we introduce the ensemble of graphs that are used over parallel channels. We present the asymptotic analysis of the performance of the corresponding codes. We then discuss the design methodology for practical systems and we present some results for VHM systems. Relevant work regarding the application of LDPC codes for parallel channels can be found in [80].

Second, we consider the construction of rate-compatible LDPC codes via puncturing, one of the most common methods used to construct rate-compatible codes. In this method, in order to change the rate of a code to a higher rate, we puncture (delete) a subset of the codeword bits. Puncturing has been studied for convolutional and turbo codes [48], [49], and [55]. The near Shannon-limit performance of LDPC codes [72] , [54], [24] motivates us to construct rate-adaptive LDPC codes. Previous work on finding puncturing patterns for LDPC codes is given in [45] where it is shown that punctured LDPC codes exhibit desirable properties. First, the performance of a good LDPC code is maintained for a wide range of rates (as defined in section 5.3, we define the performance as the ratio of the code rate to the channel capacity for small enough bit error rates). Second, there is no theoretical limitation on the number of rates or the values of rates we can generate. In Section 5.3 we present some results on punctured codes and show that a randomly punctured LDPC code usually has a good performance. We show that a punctured code can be modeled as a code that is used over two parallel channels as Fig. 14. In this model, punctured bits are transmitted over the second channel that has a zero capacity. Thus, our proposed density evolution formulas for the parallel channels can be used to find optimum puncturing patterns for the

LDPC codes. We study rate-compatible LDPC codes in more detail in chapter 6.

Third, we consider a closely related problem of unequal error protection (UEP). Some previous works on unequal error protection (UEP) codes can be found in [75], [73], [61] and [13]. In Section 5.4, we will be concerned with a possibly uniform channel; however, we would like to impose intentional non-uniform bit error rates for different sets of bits. In other words, we would like to protect some bits more than others. In particular, we are interested in unequal error correction for data frames. A transfer frame consists of a header, a body and a trailer. We usually want a smaller error probability for the header information, which contains important routing information such as the destination address and the frame number. It is also desirable to be able to read the header data without decoding the whole frame. This prevents all intermediary routers from having to decode the entire frame. Specifically, suppose we send data in the forms of frames of length n over a network. These n bits include the redundant bits due to the error correcting code. Moreover, suppose a very small fraction of the data in a frame (the header bits), consisting of $\xi(n)$ bits, is very important to us. Let us call them important bits. We need a coding scheme with the following properties. First, the important bits must have a considerably smaller error rate than the rest of the bits in the codeword. Second, for a given code rate, the average bit error rate of the code must be acceptable. In other words, we want to minimize the price that we may have to pay for the unequal error protection. Thus, we would like the UEP code to have overall performance close to the best ordinary codes for the same rate and block length. Third, we want to be able to decode the header data without decoding the whole frame. Our goal in Section 5.4 is to show that we can satisfy the above requirements with LDPC codes. The good performance of LDPC codes makes them good candidates for the problem described above.

Throughout the chapter we assume the following terminology. By a graph we mean a simple graph, i.e., a graph with no loops (edges joining a vertex to itself) and no multiple edges (several edges joining the same two vertices). Let A be a subset of the vertices in the graph g. Then $N(A) = N^1(A)$ shows the set of neighbors of A in g. More generally, for $j \in \mathbb{N}$, $N^j(A)$ is the set of vertices in g from which there is path of length j to a vertex in A.

Let D be a subgraph of g such that its vertex set is A. We say D is induced by A if D contains all edges of g that join two vertices in A. For a square matrix M, $r(M)$ denotes the spectral radius of M. In other words, $r(M) = \max\{|\eta| : \eta \text{ is an eigenvalue of } M\}$. Similar to [54], for a random variable X with distribution F_X we define $P_e(F_X) = \Pr\{X < 0\} + \frac{1}{2}\Pr\{X = 0\}$.

5.2 Non-uniform Error Correction

5.2.1 VHM Systems

Some practical applications may benefit from the use of non-uniform error protection. For example, in holographic data storage, information is recorded and retrieved in the form of two-dimensional data pages (i.e., two-dimensional patterns of bits). The bits in a page are subject to different sources of noise and interference (such as inter-page interference (IPI), limited diffraction, aberration, misalignment error and non-uniform erasure [22]). The noise distribution at any point in the page is obtained by the superposition of these noise sources. We assume that the noise is Gaussian and the signal to noise ratio (SNR) decreases as we move from the center to the corner of the page [22]. Typically, the raw bit error rate might vary by two or three orders of magnitude over a page. The common approach to solve the non-uniform error protection problem is to use an interleaver followed by a Reed-Solomon code [22]. It will be shown through simulations that LDPC codes optimized for non-uniform channels, result in an increase in the storage capacity of a typical holographic data storage by more than 50 percent compared to the approach using an interleaver and a Reed-Solomon (RS) code. In this section we discuss the design methodology for the LDPC codes that are used in the VHM systems. However, note that this design procedure is also applicable to other systems such as rate-compatible codes, OFDM systems and multi-level coding.

Consider a VHM page of $N \times N$ pixels. Each pixel is subject to noise with a probability density that is dependent on the pixel location in the page. Generally, pixels at the corner of a data page have higher probability of error than those at the center of the page. We divide this page into k_r regions in which pixels are subject to almost the same noise power. Let the regions be $R_1, R_2, ..., R_{k_r}$. All the bits in a page are written or read simultaneously.

Thus, this page can be modeled as k_r parallel binary input channels as in Fig. 14.

5.2.2 Ensemble $g(\Lambda, \rho)$

There are several ways to define the ensembles of LDPC codes suitable for non-uniform channels. We introduced such an ensemble in [104] but in this chapter we use a slightly simpler ensemble. Again, suppose we use a code of length n, and we transmit each codeword over the set of channels such that $n^{(j)}$ bits from every codeword are transmitted through the jth channel. Let $(x_1, x_2, ..., x_n)$ be a codeword. Let also $W^{(j)}$ be the set of bits in the codeword that are transmitted over the jth channel (type j bits). Thus we have $|W^{(j)}| = n^{(j)}$, where $|\cdot|$ denotes the cardinality of the set. For example, in the VHM system, $W^{(j)}$ is the set of bits in the jth region (i.e., $W^{(j)} = \{x_i : x_i \in R_j\}$). Now we define the ensemble $g(\Lambda, \rho)$ of bipartite graphs for non-uniform error protection. Let E be the set of edges in the graph and let $E^{(j)}$ be the set of edges that are incident with a variable node of type j. Also let $E_i^{(j)}$ be the set of the edges that are adjacent to the variable nodes of type j and degree i. We define

$$\lambda^{(j)}(x) = \sum \lambda_i^{(j)} x^{i-1} \tag{125}$$

where

$$\lambda_i^{(j)} = \frac{|E_i^{(j)}|}{|E^{(j)}|}. \tag{126}$$

Let $\Lambda = \{\lambda^{(j)}(x) : j = 1, ..., k_r\}$. Let also $\rho(x) = \sum \rho_i x^{i-1}$, where ρ_i is the fraction of edges connected to a check node of degree i [54]. We define the ensemble $g(\Lambda, \rho)$ as the ensemble of bipartite graphs with the degree distributions given by Λ and ρ. In other words, in the ensemble $g(\Lambda, \rho)$, variable nodes corresponding to bits of different types may have different degree distributions. In fact, we propose to design codes with the prior knowledge of which bits are transmitted over each channel. Our aim in this chapter is to show that this method has some advantages in certain applications.

5.2.3 Asymptotic Analysis

Similar to [54], we can find the density evolution formulas for the ensemble $g(\Lambda, \rho)$. Let us define

$$q^{(j)} = \frac{|E^{(j)}|}{|E|}. \tag{127}$$

Let $m_{vc}^{(l),(j)}$ denote the message that is sent from a variable node v of type j (i.e., $v \in W^{(j)}$) to its incident check node c at the lth iteration of the message passing algorithm. Let also $m_{cv}^{(l)}$ denote the message that the check node c sends to its incident variable node. Let $P_l^{(j)}$ and Q_l denote the densities of random variables $m_{vc}^{(l),(j)}$ and $m_{cv}^{(l)}$, respectively. Let also P_l' be the density of the message that is sent on a randomly chosen edge (from the variable node to the check node) at the lth iteration. Then, it can be shown that the formulas for the density evolution can be written as

$$P_l^{(j)} = P_0^{(j)} \otimes \lambda^{(j)}(Q_l) \tag{128}$$

$$P_l' = \sum q^{(j)} P_l^{(j)} \tag{129}$$

$$Q_l = \Gamma^{-1}(\rho(\Gamma(P_{l-1}'))) \tag{130}$$

where $\otimes$ denotes convolution and Γ is defined in [54] in the following way. If Z is a random variable with the distribution F_Z, then $\Gamma(F_Z)$ is defined as [54]

$$\Gamma(F_Z)(s, x) = I_{(s=0)}\Gamma_0(F_Z)(x) + I_{(s=1)}\Gamma_1(F_Z)(x) \tag{131}$$

where I is the indicator function and

$$\Gamma_0(F_Z)(x) = 1 - F_Z^-(-\ln\tanh(\frac{x}{2})),\ \Gamma_1(F_Z)(x) = F_Z(\ln\tanh(\frac{x}{2})). \tag{132}$$

Note that F_Z in (131) is the corresponding distribution for P_{l-1}'.

Let $c^{(j)}$ be the capacity of the jth binary channel in Fig. 14 and suppose that we use a randomly chosen LDPC code from the ensemble $g(\Lambda, \rho)$. Using (128),(129) and (130) we can prove the following lemma.

Lemma 13. *Suppose $c^{(j)} < 1$ for $j = 1, ..., k_r$ and $k_r < \infty$. Then, for any $i, j \in \{1, 2, ..., k_r\}$ we have $\lim_{l\to\infty} P_e(P_l^{(j)}) = 0$ if and only if $\lim_{l\to\infty} P_e(P_l^{(i)}) = 0$.*

Proof. (Sketch) Using (128) and the assumption $c^{(j)} < 1$, we conclude that to have $\lim_{l\to\infty} P_e(P_l^{(j)}) = 0$, the density Q_l should converge to a Delta function at infinity. This means that for any $i \in \{1, 2, ..., k_r\}$, we must have $\lim_{l\to\infty} P_e(P_l^{(i)}) = 0$. □

Using Lemma 13 and (128), (129) and (130) we can optimize the degree distribution of the code for the given channels. It seems that the design of good codes from the ensemble $g(\Lambda, \rho)$ is more difficult than the design of the ordinary irregular LDPC codes because of the larger number of parameters involved in the optimization. However, we will show that finding a good degree distribution for a set of parallel channels is simpler than the optimization of ordinary LDPC codes. The reason is that we can use simpler ensembles such as semi-regular ensembles (which will be defined later). In fact, the simplicity of design is one advantage of using the ensemble $g(\Lambda, \rho)$.

Most of the results for ordinary LDPC codes such as the concentration theorem, the cycle free convergence, the stability condition of [112] and [54], and the Gaussian approximation of [25] can also be generalized for the ensemble $g(\Lambda, \rho)$. The Gaussian approximation formulas for ensemble $g(\Lambda, \rho)$ are given in the Appendix. Here we give the stability condition for this ensemble. Other generalizations are straightforward. For simplicity, we derive the stability condition when all the channels in Fig. 14 are binary erasure channels with different erasure probabilities. Let ϵ_j be the erasure probability of the jth channel. Note that for this case the system of parallel channels is equivalent to a binary erasure channel with the erasure probability

$$\epsilon = \sum_{j=1}^{k_r} p_j \epsilon_j \tag{133}$$

where $p_j = \frac{n^{(j)}}{n}$. However, as we mentioned before it is better to work with the set of parallel channels instead of the derived single channel. Let $x_l^{(j)}$ be the fraction of erasure messages emitted from the variable nodes of type j in the lth iteration. Then the density

evolution formulas are

$$x_l^{(j)} = x_0^{(j)} \lambda^{(j)}(1 - \rho(1 - y_{l-1}))$$
$$y_l = \sum_{j=1}^{k_r} q^{(j)} x_l^{(j)} \tag{134}$$
$$x_0^{(j)} = \epsilon_j \text{ for } j = 1, 2, ..., k_r.$$

Let $\underline{X}_l$ be

$$\underline{X}_l = \begin{pmatrix} x_l^{(1)} \\ x_l^{(2)} \\ . \\ . \\ . \\ x_l^{(k_r)} \end{pmatrix}. \tag{135}$$

Then, the stability condition for the ensemble $g(\Lambda, \rho)$ can be stated as follows [105, 106].

Theorem 17. *Let $\epsilon_j < 1$ for $j = 1, ..., k_r$ and M be a $k_r \times k_r$ matrix whose element in the jth row and the ith column is $\alpha_{ji} = \lambda'^{(j)}(0)\rho'(1)q^{(i)}x_0^{(j)}$. Then, we have the following:*

- *If $r(M) > 1$, then there exists a strictly positive constant $\zeta = \zeta(\Lambda, \rho, \underline{X}_0)$ such that for all $l \in \mathbb{N}$ and for $j = 1, ..., k_r$, we have $x_l^{(j)} > \zeta$.*

- *If $r(M) < 1$, then there exists a strictly positive constant $\zeta = \zeta(\Lambda, \rho, \underline{X}_0)$ such that if $x_l^{(j)} \leq \zeta$ for some $l \in \mathbb{N}$ and for $j = 1, ..., k_r$, then $\lim_{l\to\infty} x_l^{(j)} = 0$ for $j = 1, ..., k_r$.*

Proof. By expanding the density evolution formula into the Taylor series at zero and neglecting high order terms, we get

$$\underline{X}_l = M\underline{X}_{l-1}. \tag{136}$$

If $\|\underline{X}_l\|$ is sufficiently small, then we have $\lim_{l\to\infty} \underline{X}_l = 0$ if and only if $\lim_{k\to\infty} M^k = 0$. This is equivalent to $r(M) < 1$. The rest of the proof is similar to the proof of the stability condition in [54]. □

We finally give an upper bound for the rate of the codes from the ensemble $g(\Lambda, \rho)$ with the maximum likelihood (ML) decoding. This bound is valid for the iterative decoding as

well. It is similar to the bound given in [101]. Let φ_i be the fraction of check nodes of degree i. Let us define $\Phi(x) = \sum_i \varphi_i x^i$. By a simple observation, we can find the following upper bound on the capacity of the LDPC codes over the BEC. The proof is similar to the one presented in [101] on the bound for uniform channels.

Theorem 18. *Consider k_r parallel binary erasure subchannels as Fig. 14 with erasure probabilities $\epsilon_1, \epsilon_2, ..., \epsilon_{k_r}$. Then, for an arbitrarily small error probability, we must have*

$$1 - R \geq \frac{\epsilon}{1 - \Phi(1 - \epsilon')} \tag{137}$$

where $\epsilon' = \sum_{j=1}^{k_r} q^{(j)} \epsilon_j$ and $q^{(j)}$ and ϵ are given by (127) and (133).

5.2.4 Advantages of the Ensemble $g(\Lambda, \rho)$

Here we briefly explain the advantages of using the ensemble $g(\Lambda, \rho)$. These advantages are further explained and verified using simulations in Section 5.5. Note that in our model, we know which subsets of bits are transmitted through each channel. The important fact about the ensemble $g(\Lambda, \rho)$ is that we use this information in the code design. Note that in ordinary ensembles of LDPC codes, we do not use this information in the code design, instead we use the average density of the LLR's of channels for each bit. This extra information results in several advantages of the ensemble $g(\Lambda, \rho)$ over the ordinary ensembles. The first advantage is that we can use lower values for variable nodes in the degree distribution. In other words, we can obtain sparser codes using the ensemble $g(\Lambda, \rho)$ having the same performance of ordinary LDPC codes. This results in faster decoding and more efficient implementation.

In ordinary LDPC codes ensembles, to approach the channel capacity, we need to have a high number of degree-two variable nodes in the graph [120]. Thus capacity-approaching LDPC codes usually suffer from the error floor problem. However, since we use more information in the code design of the ensemble $g(\Lambda, \rho)$, we can have codes with a low number or even no degree-two variable nodes that still have thresholds close to the Shannon limit. This is particularly very important in data storage systems such as holographic memories because a very low error probability is required.

Another advantage is simpler design. It is worth noting that in ordinary LDPC codes, regular ensembles usually do not have thresholds close to the Shannon limit. Thus, in ordinary LDPC codes in order to approach channel capacity we need to use highly irregular codes. However, in the ensemble $g(\Lambda, \rho)$ part of the required irregularity is achieved by channel non-uniformity. In fact, we will show in Section 5.5 that we can approach the channel capacity by using semi-regular codes (codes in which bits that are transmitted through the same channel correspond to variable nodes with the same degrees). This will simplify the degree optimization significantly.

Finally, we can improve the performance of the short-length codes, by using the ensemble $g(\Lambda, \rho)$ becuse more information is available in the code design. Note that ensemble $g(\Lambda, \rho)$ is a generalization of the ordinary ensembles of LDPC codes. In fact by choosing all $\lambda^{(j)}(x)$ equivalent we obtain an ordinary ensemble of LDPC codes. Thus, in all circumstances, the performance of the codes obtained from the ensemble $g(\Lambda, \rho)$ is at least as good as the codes obtained from ordinary ensembles.

5.3 Rate-Compatible LDPC Codes

In this section we are concerned with punctured codes over binary-input output-symmetric memoryless (BIOSM) channels. We use our developments for non-uniform channels to study rate-compatible LDPC codes. We restrict ourselves to normalized channels [24]. A normalized channel is defined as the channel obtained by concatenation of a BIOSM channel with log likelihood mappings. The normalization of a channel is a lossless process because the set of log likelihoods is a sufficient statistic for decoding. Thus we say two channels C_1 and C_2 are equivalent if their normalized channels are the same. We represent the capacities of the channels by c_1 and c_2, respectively. We first prove the following lemma that is useful for modeling of punctured codes.

Lemma 14. *A normalized BIOSM channel has zero capacity if and only if the received LLR is equal to zero with probability one.*

Proof. Let X and Y be the random variables representing the input and output of a BIOSM channel, respectively. We define the random variable U in the following way. We let the

input to the channel be $X = 1$. If y is the output of the channel, then

$$U = \log \frac{\text{pr}\{X = 1|Y = y\}}{\text{pr}\{X = -1|Y = y\}}, \tag{138}$$

Then the capacity of the normalized channel is given by [24]

$$c = 1 - E[\log_2(1 + e^{-U})|X = 1]. \tag{139}$$

Thus if $U = 0$ with probability one then $c = 0$. Moreover, if $c = 0$, we have $E[\log_2(1 + e^{-U})] = 1$. If we assume $p(u)$ is the probability density function of U, we have

$$\begin{aligned} 1 &= \int_{-\infty}^{+\infty} \log_2(1 + e^{-u})p(u)du \\ &= \text{pr}\{U = 0\} + \int_{(0,+\infty)} [\log_2(1 + e^{-u}) - \log_2(1 + e^{u})e^{-u}]p(u)du. \end{aligned}$$

where we used $p(u) = e^u p(-u)$ [54]. Since we have $\text{pr}\{U = 0\} \leq 1$ and $[\log_2(1 + e^{-u}) - \log_2(1 + e^u)e^{-u}] < 0$ for $u \in (0, \infty)$, we conclude $\text{pr}\{U = 0\} = 1$. □

We would like to design rate-adaptive LDPC codes that use the same encoder and decoder for all rates. Let $\mathfrak{R} = \{r_1, r_2, ..., r_s\}$ be the set of different rates that are needed. Let r_p be the rate of the parent code (i.e., the lowest rate in $\mathfrak{R}$). We consider the following scheme. We design an optimized LDPC code of rate $r_p = k/n$ where k and n are the lengths of information blocks and the codewords, respectively. To generate a code with a new rate, we find an optimum puncturing of a subset of bits in the codeword and send the punctured codeword to the receiver. It is assumed that the decoder knows the positions of the punctured bits in the codeword. At the beginning of the iterative decoding, we need to compute log likelihood ratios (LLR's) in the decoder. The LLR's for the punctured bits are set to zero.

Let us define the performance of a rate-compatible code over a channel as $\frac{r}{c}$ where c is the channel capacity and r is the maximum rate of the code for which the error probability is less than a required value. When we consider asymptotic behavior of codes, r is the maximum rate for which an arbitrarily small error probability is achievable. Now consider

a time varying binary-input output-symmetric channel [54] which can be described by its transmission conditional probability $P(y|x,\alpha)$ where $\alpha \in [\alpha_{min}, \alpha_{max}]$ is time variant. For example, for binary-input additive white Gaussian noise (BIAWGN) channels, α can be the variance of the noise. Let $c(\alpha)$ be the capacity of this channel. We may assume that $c(\alpha)$ is a decreasing function of α. We design an optimal LDPC code for the rate r_p that is used when $\alpha = \alpha_{max}$. Now, suppose the channel quality improves. In other words, the value of the parameter α is reduced to a value less than α_{max}. By puncturing, we increase the code rate from r_p to $r(\alpha)$ such that the error probability still becomes less than the required value. If $\frac{r_p}{c(\alpha_{max})} > \frac{r(\alpha)}{c(\alpha)}$ then we would have a performance loss due to puncturing. Our goal is to minimize the performance loss by finding a good puncturing pattern.

To investigate the performance of punctured LDPC codes, we consider the model depicted in Fig. 27. In this model it is assumed that the unpunctured bits are transmitted through the channel and the punctured bits are transmitted through a virtual channel with a zero capacity. In fact, by Lemma 14, a normalized BIOSM channel with zero capacity is equivalent to a BEC with erasure probability one. Let p be the fraction of punctured bits and define $r_{eq}(\alpha)$ to be the code rate of the overall channel in Fig. 27. In other words, $r_{eq}(\alpha)$ is the code rate if we consider both punctured and unpunctured bits. With this definition, it is clear that $r_{eq}(\alpha) = r_p$. Note that $C_{eq} = C_{eq}(\alpha)$ is the channel that consists of two subchannels C_1 and C_2 with capacities $c(\alpha)$ and zero, respectively. Therefore, a fraction p of bits are transmitted through C_2 and the rest of the bits are transmitted through C_1. Let $c_{eq}(\alpha)$ be the capacity of $C_{eq}(\alpha)$. In Fig. 27 we have

$$c_{eq}(\alpha) = (1-p)c(\alpha), \quad r(\alpha) = \frac{r_p}{(1-p)}. \tag{140}$$

Therefore, we have a performance loss due to puncturing, if and only if $c_{eq}(\alpha) > c(\alpha_{max})$. Let z denote the LLR of the received bits and $\varpi(z;\alpha)$ be the density of z when the all-zero codeword is sent. Then by the following theorem, we identify the channels for which the code performance does not change due to random puncturing. For example, as a special case of the following theorem, we conclude that for a binary erasure channel in which α is chosen to be the erasure probability, a random puncturing results in no performance

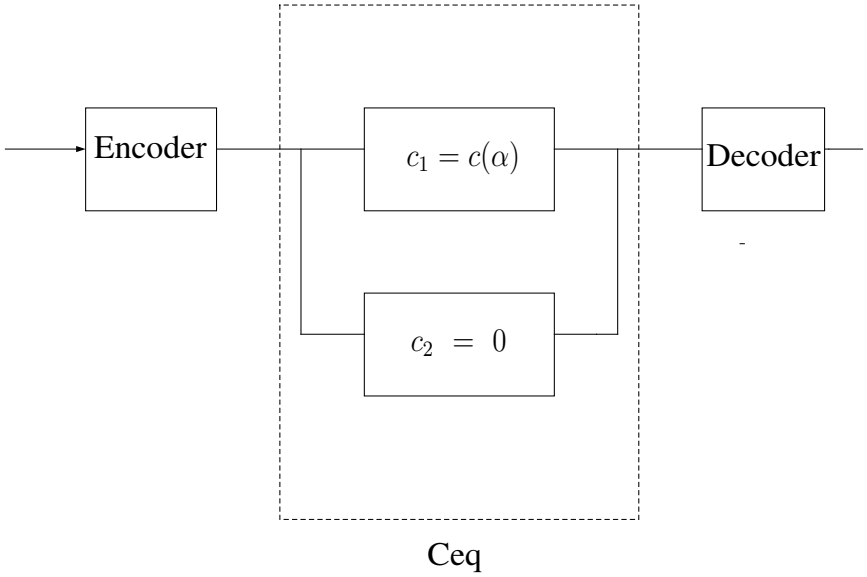

Figure 27: A model that describes puncturing over a binary channel.

loss. In fact, the performance of the randomly punctured code is the same for all rates. It is important to note that for other types of channels, we usually have some performance degradation because of puncturing. Therefore, we need to optimize the puncturing pattern for these types of channels.

Theorem 19. *[106] Let* $\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z)$ *specify a normalized channel in which* θ *is an increasing function of* α *such that for all* $\alpha \in [\alpha_{min}, \alpha_{max}]$*, we have* $0 \leq \theta(\alpha) \leq \theta(\alpha_{max}) \leq 1 - r_p$ *and* $\int_{-\infty}^{+\infty} f(z)dz = 1$*. Then, the average performance of any binary block code does not change by random puncturing if we choose the puncturing fraction* $p(\alpha)$ *properly. Moreover, the class of channels defined by* $\varpi(z;\alpha)$ *is the only class of normalized BIOSM channels having this property.*

Proof. First we prove the following lemma:

Lemma 15. *The performance of an arbitrary block code with an arbitrary decoder does not change by random puncturing in the scheme of Fig. 27 if and only if there exists a puncturing fraction function* $p(\alpha)$ *such that for all* $\alpha \in [\alpha_{min}, \alpha_{max}]$*, we have* $c_{eq}(\alpha) \equiv c(\alpha_{max})$

Proof. Suppose the error probability of all decoders in Fig. 27 stays the same for any random puncturing. Then, the probability density function of the input of the decoders

must remain unchanged by puncturing. This implies that $C_{eq} \equiv C(\alpha_{max})$. Moreover, suppose there exists a puncturing fraction function $p(\alpha)$ such that for all $\alpha \in [\alpha_{min}, \alpha_{max}]$ we have $C_{eq}(\alpha) \equiv C(\alpha_{max})$. Then if we perform random puncturing according to $p(\alpha)$, we have

$$\frac{r(\alpha)}{c(\alpha)} = \frac{\frac{r_p}{1-p}}{\frac{c_{eq}(\alpha)}{1-p}} = \frac{r_p}{c(\alpha_{max})}. \tag{141}$$

Therefore, the performance of the code stays the same. □

Proof of Theorem 19: Suppose the assumptions of the theorem hold for $\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1-\theta(\alpha))f(z)$. We choose

$$p(\alpha) = \frac{\theta(\alpha_{max}) - \theta(\alpha)}{1-\theta(\alpha)}. \tag{142}$$

It is clear that we have $0 \leq\ p(\alpha) \leq \theta(\alpha_{max}) \leq 1 - r_p$. The LLR corresponding to $C_{eq}(\alpha)$ is equal to

$$\begin{aligned} &(1-p(\alpha))\varpi(z;\alpha) + p(\alpha)\delta(z) \\ &= \theta(\alpha_{max})\delta(z) + (1-\theta(\alpha_{max}))f(z) = \varpi(z;\alpha_{max}). \end{aligned} \tag{143}$$

This is the same as the LLR for $C(\alpha_{max})$. Therefore, we have $C_{eq}(\alpha) \equiv C(\alpha_{max})$. By Lemma 15, we conclude that the performance of the codes on this channel does not change by random puncturing.

Now suppose we have a time varying channel that is defined by $\varpi(z;\alpha)$ such that the performance of codes stays the same by random puncturing. By Lemma 15, we have $C_{eq}(\alpha) \equiv C(\alpha_{max})$. Therefore, we conclude that $\varpi(z;\alpha)(1-p(\alpha)) + p(\alpha)\delta(z) = \varpi(z;\alpha_{max})$. This results in

$$\varpi(z;\alpha) = \frac{\varpi(z;\alpha_{max})}{1-p(\alpha)} - \frac{p(\alpha)\delta(z)}{1-p(\alpha)}. \tag{144}$$

Let p_m be the maximum possible fraction of punctured bits. It is clear that $p_m \leq 1 - r_p$. Let us define $\theta(\alpha_{max}) = p_m$ and

$$\begin{aligned} \theta(\alpha) &= \frac{\theta(\alpha_{max}) - p(\alpha)}{1-p(\alpha)}, \\ f(z) &= \frac{\varpi(z;\alpha_{max}) - \theta(\alpha_{max})\delta(z)}{1-\theta(\alpha_{max})}. \end{aligned} \tag{145}$$

Since $p(\alpha)$ is decreasing in α and $0 \leq p(\alpha) \leq \theta(\alpha_{max}) \leq 1 - r_p$, we conclude that θ is an increasing function of α and $0 \leq \theta(\alpha) \leq \theta(\alpha_{max}) \leq 1 - r_p$. Moreover, we have $\varpi(z;\alpha) = \theta(\alpha)\delta(z) + (1 - \theta(\alpha))f(z)$ and $\int_{-\infty}^{+\infty} f(z)dz = 1$.

□

It is shown in [24] that if we optimize an LDPC code for a symmetric channel, the code usually has good performance on other types of symmetric channels for which the code is not optimized. This property of LDPC codes can be used to explain the good performance of punctured LDPC codes by examining Fig. 27 as follows. The figure implies that the puncturing process can be considered as a change in the channel instead of the change in the code rate. Therefore, although the LDPC code is optimized for the channel with the parameter $\alpha = \alpha_{max}$ (or $p = 0$) we expect that it also performs well for other values of α for which $p > 0$ (note that C_{eq} is a symmetric channel). However, we can optimize the puncturing pattern to further improve the performance.

Considering Fig. 27, we can find the density evolution formulas for a punctured LDPC code over a BIOSM channel using the density evolution formulas for the ensemble $g(\Lambda, \rho)$. Then, using these formulas, we obtain good puncturing distributions for LDPC codes. If the channel is subject to Gaussian noise we can also apply the Gaussian approximation method. We now show that by applying the Gaussian approximation formulas of Appendix B we get the same result as [45].

Let $m_u(l)$ denote the mean of the messages from check nodes to variable nodes in the lth iteration. Let also $m_0 = \frac{2}{\sigma^2}$ where σ is the variance of C_1 in Fig. 27. We define $\psi_i^{(1)}$ to be the fraction of unpunctured variable nodes of degree i among all the unpunctured variable nodes in the graph. Define $\psi_i^{(2)}$ for the punctured variable nodes similarly. If the puncturing fraction is p, we have

$$\psi_i = (1 - p)\psi_i^{(1)} + p\psi_i^{(2)}, \tag{146}$$

$$\sum_i \psi_i = \sum_i \psi_i^{(1)} = \sum_i \psi_i^{(2)} = 1. \tag{147}$$

Our goal is to find $\{\psi_i^{(2)}\}_{i>1}$ such that the performance of the code is optimized. We

have

$$\lambda_i^{(j)} = \frac{i\psi_i^{(j)}}{\sum\limits_k k\psi_k^{(j)}}, \quad j = 1, 2. \tag{148}$$

Let $p_{pe} = q^{(2)} = \frac{|E^{(2)}|}{|E|}$. Using the Appendix we define

$$h_i^{(1)}(s, r) = \phi\Big(s + (i-1)\sum_j \rho_j \phi^{-1}(1 - (1-r)^{(j-1)})\Big), \tag{149}$$

$$h_i^{(2)}(s, r) = \phi\Big((i-1)\sum_j \rho_j \phi^{-1}(1 - (1-r)^{(j-1)})\Big), \tag{150}$$

$$h(s, r) = (1 - p_{pe})\sum_i \lambda_i^{(1)} h_i^{(1)}(s, r) + p_{pe}\sum_i \lambda_i^{(2)} h_i^{(2)}(s, r). \tag{151}$$

Then we have

$$r_l = h(s, r_{l-1}) \tag{152}$$

where $s = m_0$ and $r_0 = (1 - p_{pe})\phi(s) + p_{pe}$. As it is stated in the Appendix, $r_l(s) \longrightarrow 0$ if and only if $r > h(s, r)$ for all $r \in (0, 1)$. We now set up a linear program to optimize the puncturing pattern. Let us define $\mu_i = (1 - p_{pe})\lambda_i^{(1)}$ and $\beta_i = p_{pe}\lambda_i^{(2)}$. Here we maximize p for the given α. Thus we have the following optimization problem

$$\max_{\mu_i, \beta_i} p = \frac{|E|}{n}\sum_i \frac{\beta_i}{i} \tag{153}$$

with the constraints

$$h(s, r) = \sum_i \mu_i h_i^{(1)}(s, r) + \sum_i \beta_i h_i^{(2)}(s, r) < r, \quad 0 < r < 1 \tag{154}$$

$$\mu_i + \beta_i = \lambda_i. \tag{155}$$

After finding the optimum values of β_i and μ_i, we can find p_i (the fraction of the variable nodes of degree i that should be punctured) by the following equations:

$$\begin{aligned} P_{pe} &= \sum_i \beta_i, \\ \lambda_i^{(2)} &= \frac{\beta_i}{p_{pe}}, \\ \psi_i^{(2)} &= \frac{\frac{\lambda_i^{(2)}}{i}}{\sum\limits_k \frac{\lambda_k^{(2)}}{k}}, \\ p_i &= \frac{p\psi_i^{(2)}}{\psi_i}. \end{aligned} \tag{156}$$

We note that this is the same as the result in [45].

5.4 Unequal Error Protection Using LDPC Codes

We now consider a problem closely related to code design for the non-uniform channels. We are concerned with uniform channels; however, we would like to impose intentional non-uniformity at the bit error rates of different sets of bits. In other words, we would like to protect some bits more than others. In particular, we are interested in unequal error correction for data frames. A transfer frame consists of a header and a body. The header length is usually small compared to the body. In fact, it has usually logarithmic length with respect to the frame length [56]. Thus if $\xi(n)$ is the header length, it is reasonable to assume $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$. We usually want a very small error probability for the header information.

Suppose we want to transmit a block of k bits over a BIOSM channel. We also want to use at most $n-k$ redundant bits. Let $\xi(n)$ be the number of important bits that require higher protection. One approach is to use two different block codes, one for the important bits and the other for the rest of the bits. However, it is more interesting to design only one block code that provides unequal error protection. More importantly, using two different LDPC codes is not efficient for the following reason. Since $\xi(n)$ is usually a very small number, we have to use a short-length code for the important bits. However, as we know LDPC codes do not perform well for short lengths. Therefore to get a good bit error rate we must use a very low-rate code which is inefficient. Thus we need to use a different type of code for the important bits. On the other hand, it is not clear that using only one LDPC code and imposing the unequal error protection on the code would result in an efficient coding scheme. In fact, our aim in this section is to study this.

5.4.1 Perfect Protection

Suppose we transmit binary bits over a binary channel with capacity $c(\alpha)$ where α is the parameter of the channel. We want to use a block code of rate R that performs unequal error protection. Let $P_E(C,\alpha)$ be the average error rate of the code C when the channel parameter is equal to α. Let also $P_E^{\xi}(C,\alpha)$ be the average error rate of the important bits.

Let $\mathcal{C}_R$ be the class of codes of type $\mathcal{C}$ and rate R. For example $LDPC_R$ is the class of LDPC codes of rate R. We define $\mathcal{C}_R^{\epsilon,\delta}$ as the class of unequal error protection codes of type $\mathcal{C}$ and rate R that satisfy the following property. For any $C \in \mathcal{C}_R^{\epsilon,\delta}$ if $c(\alpha) > \delta$, then we have $P_E^{\xi}(C, \alpha) < \epsilon$.

Definition 2. *We say that an unequal error protection scheme $\mathcal{C}_R^{\epsilon,\delta}$ perfectly protects the important bits if for any positive numbers ϵ and δ and any code C in $\mathcal{C}_R$, there exists a code C' in $\mathcal{C}_R^{\epsilon,\delta}$ such that $P_E(C', \alpha) \leq P_E(C, \alpha)$.*

Intuitively, perfect protection implies an unequal error protection without paying any price. In other words, even if the channel capacity $c(\alpha)$ becomes arbitrary close to zero, we are able to get arbitrarily small error probability for the important bits without loosing anything with respect to other bits. It can be seen that for asymptotically good codes, perfect protection is possible only when $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$, otherwise we violate the fundamental theorem of Shannon capacity. This assumption is reasonable for applications such as data frames where $\xi(n) << n$.

5.4.2 An Unequal Error Protection Scheme

Now we propose a scheme for unequal error protection using LDPC codes. Conventional LDPC codes provide almost equal error protection. Although high-degree variable nodes have lower error probabilities in irregular LDPC codes, the difference between the error rates of the variable nodes of different degrees is not considerable (usually less than one order of magnitude). Moreover, this difference reduces when the channel becomes worse. Note that it is usually difficult if not impossible to find good unequal error protection LDPC codes by searching for different degree distributions. This is because we have to choose the degree distribution to be extremely irregular, (i.e., we have to choose very high degrees for the important bits) which is usually harmful if we cannot have a large enough code length.

Let A be the set of important variable nodes and $|A| = \xi(n)$. We propose a scheme based on the degree distributions of the vertices in the sets $N^0(A) = A$, $N(A)$, $N^2(A)$, ..., $N^h(A)$, where h is a constant. Note that $N^j(A)$ consists of variable nodes if j is even. Otherwise, $N^j(A)$ consists of check nodes. As it is explained in [71], from the point of view

of variable nodes, it is best to have high degrees. On the contrary, for check nodes, it is best to have low degrees. In fact, our scheme is based on the above fact. Let $g_n(\lambda, \rho)$ be the ensemble of irregular graphs introduced in [54] and [70]. That is the ensemble of bipartite graphs having degree distribution (λ, ρ) and length n. We define $g_n(\lambda, \rho, h, d_v, d_c)$ as the ensemble of bipartite graphs for which the degree of each vertex in $N^j(A)$ for $j = 0, 1, .., h$ is equal to d_v if j is even. Otherwise, it is equal to d_c. The degree distribution of the vertices in the rest of the graph is determined by λ and ρ similar to the ensemble $g_n(\lambda, \rho)$. Note that for simplicity we assume that the degrees of all the variable nodes in the sets $N^0(A) = A$, $N(A)$, $N^2(A)$, ..., $N^h(A)$ are the same and the degrees of all the check nodes in these sets are the same. We could have also assigned an irregular degree distribution to the vertices in $N^j(A)$. In general, a graph from the ensemble $g_n(\lambda, \rho, h, d_v, d_c)$ is similar to a graph from the ensemble $g_n(\lambda, \rho)$ having the extra condition that the vertices of A and their neighborhood of depth h must have certain degree distributions. We have the following theorem [106].

Theorem 20. *If* $\lim_{n\to\infty} \frac{\xi(n)}{n} = 0$*, then the ensemble of the codes defined by* $g_n(\lambda, \rho, h, d_v, d_c)$ *satisfies the perfect protection property. In other words, for any positive numbers* ϵ *and* δ *and any code* C *in* $g_n(\lambda, \rho)$ *of rate* R*, there exists a code* C' *in* $g_{n'}(\lambda, \rho, h, d_v, d_c)$ *having the same rate as* C *such that* $P_E(C', \alpha) \leq P_E(C, \alpha)$*. Furthermore, if* $c(\alpha) > \delta$ *we have* $P_E^{\xi}(C', \alpha) < \epsilon$*.*

Proof. For simplicity we prove the theorem for the binary erasure channel. The extension to other channels is immediate. Let α be the erasure probability. Thus $c(\alpha) = 1 - \alpha$. Suppose we are given positive numbers ϵ and δ and a degree distribution pair (λ, ρ) of rate R. We show that for sufficiently large n', the ensemble $g_{n'}(\lambda, \rho, h, d_v, d_c)$ satisfies the requirements of the theorem. Define

$$B = \bigcup_{j=0}^{h} N^j(A). \tag{157}$$

Since $|B| \leq \max(d_v, d_c)^h \xi(n)$ we have $\lim_{n\to\infty} \frac{|B|}{n} = 0$. Let x_l be the average fraction of erasure messages which are passed in the lth iteration of the iterative decoding on a graph g in $g_n(\lambda, \rho)$. Let also y_l be the average fraction of erasure messages which are passed in

the lth iteration of the iterative decoding on a graph g' in $\mathcal{C}_{n'}(\lambda, \rho)$. As $n' \to \infty$ if we pick a vertex from g' at random, with high probability, its neighborhood of depth d does not contain any vertices from the set B for any constant d. Thus by arguments similar to the ones in [112], we conclude that $y_l \to x_l$ as $n, n' \to \infty$. Therefore, for sufficiently large n', we have $P_E(C', \alpha) \leq P_E(C, \alpha)$.

Now let I be the subgraph of g' that is induced by the vertices in B as shown in Fig. 28. Let t_l be the probability that the value of a variable node in A, the set of important bits, is unknown after the lth iteration. Note that with high probability the graph I is cycle-free. Thus using the similar arguments as in [112] and [70], we conclude that for $l \leq h$, t_l satisfies

$$\begin{aligned} z_0 &= \alpha \\ z_l &= z_0\big[1 - (1 - z_{l-1})^{d_c - 1}\big]^{d_v - 1} \\ t_l &= z_0\big[1 - (1 - z_{l-1})^{d_c - 1}\big]^{d_v}. \end{aligned} \tag{158}$$

Using (158), we see that for any $0 \leq \alpha < 1$ we can always choose the parameters h, d_v, and d_c such that $t_h < \epsilon$. Hence, we have $P_E^{\xi}(C', \alpha) < \epsilon$.

□

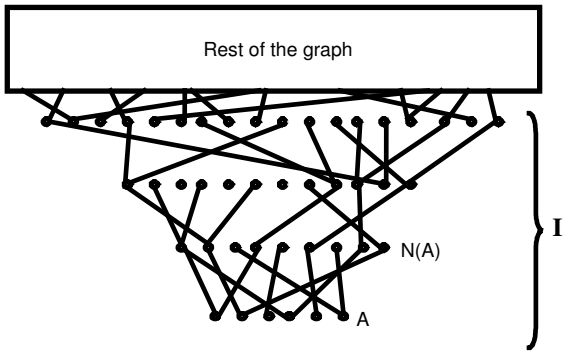

Figure 28: Illustration of the subgraph I.

Theorem 20 does not guarantee that the proposed scheme is efficient for short-length codes. However, it gives some ideas how to design short-length codes. Our simulations suggest that the scheme also results in good performance for short-length codes.

5.4.3 Decoding of Highly Protected Bits

As we mentioned previously, it is desirable in some applications that we can decode the important bits without having to decode the entire received word. This is particularly interesting in network applications where the header has to be protected more than the rest of the bits and extracted in routers. Here, we show that this is possible using the proposed unequal error-protection LDPC (UELDPC) codes. The key point is that by the proof of Theorem 20, we conclude that only h iterations are sufficient to obtain a small enough error probability for the important bits. Note that we only need the messages sent to the important bits at the hth iteration (note that h is an even number). Thus, for the decoding we only need the subgraph I in Fig. 28.

To decode the important bits we perform the following procedure. First, the variable nodes in $N^h(A)$ send messages to the check nodes in $N^{h-1}(A)$. These messages are simply the LLR's of the variable nodes based on the observation of the channel. Then, the check nodes in $N^{h-1}(A)$ send messages to the variable nodes in $N^{h-2}(A)$. These messages are computed based on the messages from $N^{h-1}(A)$. We continue until the messages to $N^0(A) = A$ are computed. Thus, we need to compute $|E(I)|$ messages for decoding the important bits ($|E(I)|$ is the number of the edges of the graph I). Note that the number of messages that must be computed for decoding the entire block is equal to $|E| \times 2l$, where l is the total number of iterations in the message passing algorithm and $|E|$ is the total number of edges in the Tanner graph of the code. Let T_ξ and T be the amount of time required for decoding the important bits and the whole block, respectively. Then we have

$$\frac{T_\xi}{T} = \frac{|E(I)|}{|E| \times 2l} \to 0 \quad \text{as } n \to \infty. \tag{159}$$

In fact $T_\xi = \Theta(\xi(n)) = o(n)$ but $T = \Theta(n)$. Therefore, we conclude that the important bits can be decoded in a much shorter time than the time required for decoding the entire block.

5.5 Practical Code Design and Simulation Results

5.5.1 Practical Code Design for Non-Uniform Channels

We now consider the problem of designing efficient LDPC codes from the ensemble $g(\Lambda, \rho)$ for the VHM systems. It is known that long LDPC codes can have performance close to the Shannon limit. The use of long LPDC codes is possible in the VHM systems because the whole memory page is read or stored simultaneously. Since bit error rates of less than 10^{-12} is desirable for the VHM systems, we require that the code do not present an error floor at least for the bit error rates (BERs) of higher than 10^{-12}.

A stopping set S is defined in [27] as a subset of variable nodes such that all neighbors of S are connected to S at least twice. It is shown in [28] and [109] that if $\lambda_2\rho'(1) < 1$ (λ_2 is the fraction of the edges connected to the variable nodes of degree two), then the minimum distance and the size of the minimum stopping set in the expurgated ensemble increase linearly with respect to the code length. Here, a constant fraction of the codes in the ensemble with low minimum stopping set size are removed in the expurgation. On the other hand, if $\lambda_2\rho'(1) > 1$, these quantities are sublinear with high probability. Until now, all the discovered capacity-achieving sequences of LDPC codes over the BEC, satisfy $\lambda_2\rho'(1) > 1$ [84]. Therefore, for achieving the capacity, we should have a small minimum distance [28]. This implies that capacity achieving codes have the error floor effect. In fact, capacity-approaching codes of practical lengths usually have an error floor at the BER of 10^{-7} or higher. On the other hand, if the minimum distance is linear, the error-floor effect is reduced substantially. Although, we do not have a rigorous proof for this, simulations show the superiority of these codes in terms of the error-floor effect over the codes with sublinear minimum distance. Thus, in our designs we always use the expurgated ensembles with linear minimum distance and linear minimum stopping set size. Now, we discuss the code design for the non-uniform error correction. For simplicity, we first consider the binary erasure channel. The VHM systems in which we have BIAWGN channels will be discussed afterwards.

5.5.1.1 Binary Erasure Channel

Consider the case that all the channels in Fig. 14 are binary erasure channels having different erasure probabilities. Let ϵ_j be the erasure probability of the jth channel. Here, we compare the performance of ordinary irregular LDPC codes and the codes from the ensemble $g(\Lambda, \rho)$.

As an example, consider the case that the number of channels $k_r = 4$ and

$$\epsilon_1 = .1\kappa, \quad \epsilon_2 = .25\kappa, \quad \epsilon_3 = .5\kappa, \quad \epsilon_4 = .95\kappa, \tag{160}$$

where κ is a constant. Suppose we use half-rate LDPC codes of length 10^4 in which 2500 bits are transmitted over each channel. Note that the whole system can be modeled as a binary erasure channel with the erasure probability $\epsilon = .45\kappa$. As a first approach, we consider the performance of the optimized half-rate LDPC codes for the erasure channel in [1]. We also design codes using the ensemble $g(\Lambda, \rho)$ as follows. In our design to alleviate the error floor problem we require that $\lambda_2 \rho'(1) < 1$. For design simplicity, we choose the degree distribution to be semi-regular. By a semi-regular degree distribution, we mean a degree distribution in which the variable nodes of the same type (variable nodes corresponding to bits that are transmitted through the same subchannel) have the same degree. We also require that the degree distribution of the check nodes be concentrated at two consecutive values. It is observed that this limitation does not result in considerable performance loss. However, it makes the optimization simpler [24] and [54]. We denote the ensemble of semi-regular codes by $g(D, \rho)$ where $D = \{d_j : j = 1, ..., k_r\}$ and d_j is the degree of the variable nodes of type j. Thus, for the above example, a semi-regular degree distribution consists of at most four distinct degrees for the variable nodes. It may sound that the semi-regularity is too restrictive and the performance of the resulting codes would be much worse than the fully optimized codes. However, this is not the case. For the length $n = 10^4$, the best half-rate ordinary irregular code that we found in [1] has the following degree distribution

$$\lambda_1(x) = 0.2498x + .2472x^2 + .1480x^5 + .0033x^6 + .3517x^{19},$$
$$\rho_1(x) = x^7$$

Let Code A be a randomly chosen code of length 10^4 from the ensemble defined by (λ_1, ρ_1). Note that the maximum variable-node degree is 20 for this code. We now design a semi-regular code from the ensemble $g(D, \rho)$. To simplify the design we restrict the maximum variable-node degree to be 7. Therefore, only a few choices are left. We can easily find the best possible code with the given constraints using density evolution. For example, we found the following degree distribution

$$d_1 = 4, d_2 = 7, d_3 = 3, d_4 = 2, d_c = 8. \tag{161}$$

Let Code B be a randomly chosen code of length 10^4 from the ensemble that is defined by the above degree distribution. Since the maximum variable-node degree in Code B is 7, we also generated the best code (with respect to threshold) given in [1] with the maximum variable-node degree 7. Let Code C be a randomly chosen code of length 10^4 from this ensemble. Fig. 29 shows the performance of these codes. First, we note that both of the codes A and C have an error floor higher than 10^{-5} while Code B does not have any error floor at least for the BERs higher than 10^{-9}. Furthermore, for almost all practical purposes, Code B is the best among these codes. It is worth noting that the maximum variable-node degree of Code A is much higher than Code B. We also conclude that Code B has lower bit error rates than the Code C for all values of ϵ, the average channel erasure probability. The performance of Code B over one single binary erasure channel is also shown in the figure. We observe that the performance of the code over the non-uniform channel (4 parallel subchannels) is much better than its performance over the equivalent single channel. This verifies that we have utilized of the non-uniformity of the channel in the code design. Additionally, if it is desired, by slightly relaxing the constraints on the ensemble $g(D, \rho)$, we can get closer performance to the capacity.

Asymptotically, the probability of a small (logarithmic size) stopping set in the expurgated ensemble that we defined in above (Code B) goes to zero . Since we are concerned with the error floor, we need to be careful about variable nodes of degree two. In fact, any cycle whose variable nodes have degree two constructs a stopping set. Thus, if one of these cycles exists in our code, we just regenerate the code. Since the probability of having these

small stopping sets is bounded away from one, it is very likely that we get a code with no small stopping set by a few trials. It is worth noting that these cycles can be found using simple graph algorithms [26]. Therefore, we avoid degree-two variable nodes in our design at the next section. We will show that we can still find very good codes from the ensemble $g(D, \rho)$.

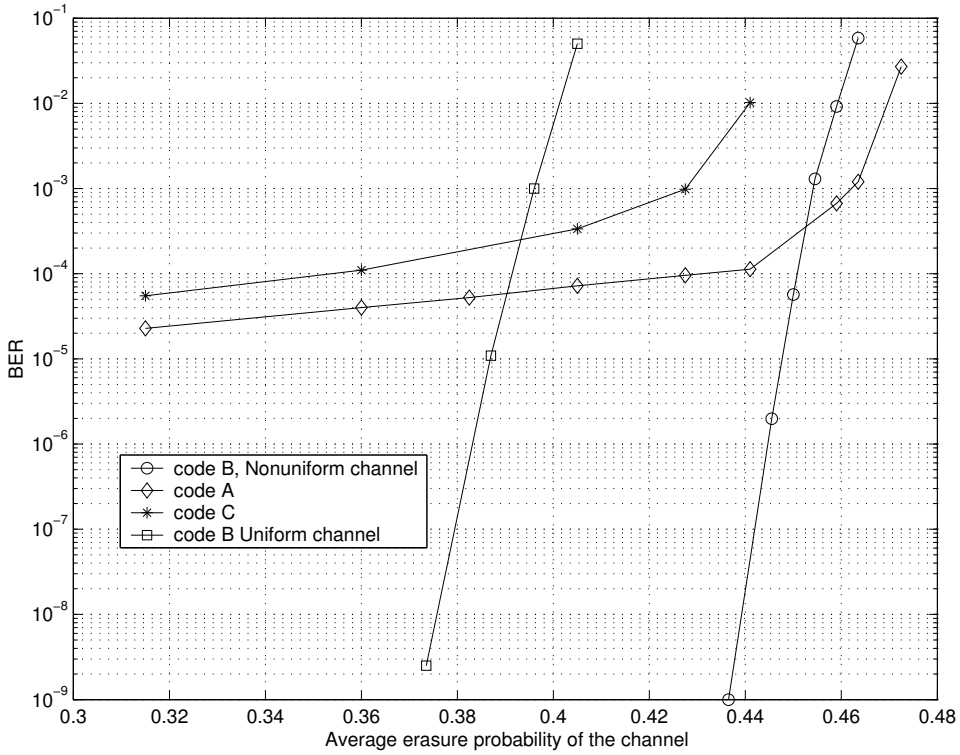

Figure 29: Performance of different half-rate LDPC codes over the BEC.

5.5.1.2 Binary Input Additive White Gaussian Noise Channel

We now consider the VHM systems. As we mentioned before, each page of the VHM system can be considered as a set of parallel channels having different noise powers. As an example, we use the VHM system in [107]. In this system we divide each page into four regions ($k_r = 4$). The noise is assumed to be Gaussian. Therefore, the system can be modeled as a set of four binary-input additive white Gaussian noise (BIAWGN) channels.

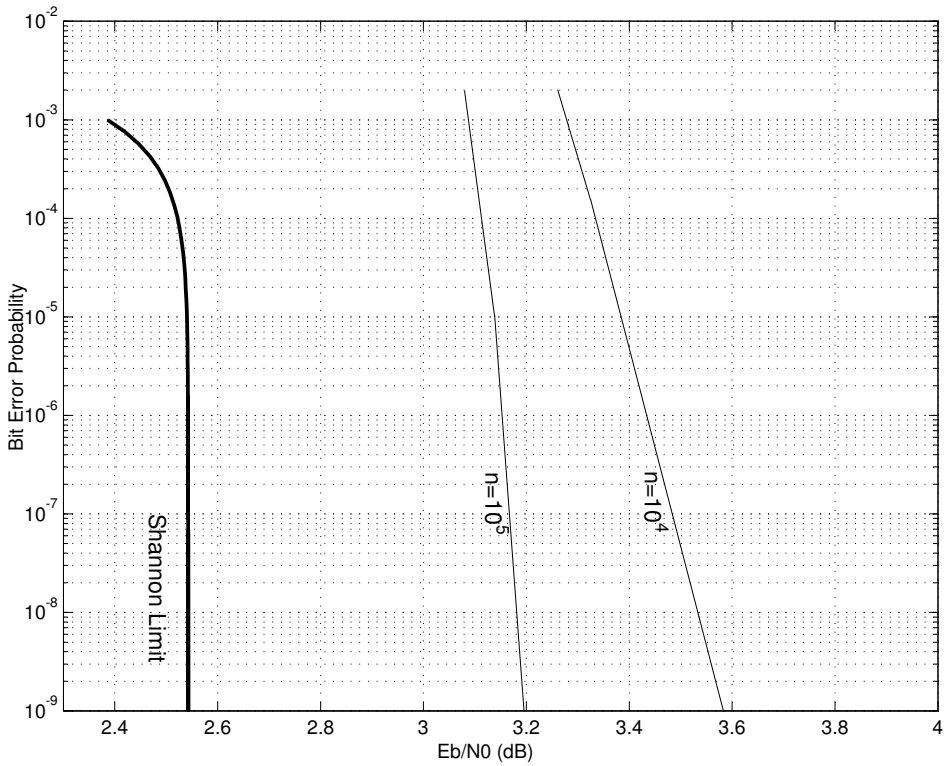

Figure 30: Performance of the irregular LDPC code of rate .85 over four parallel BIAWGN channels.

The relative SNR's of different regions are

$$\begin{aligned} SNR_2 - SNR_1 &= 1.61 \text{ dB} \\ SNR_3 - SNR_1 &= 2.80 \text{ dB} \\ SNR_4 - SNR_1 &= 3.74 \text{ dB}. \end{aligned} \tag{162}$$

We want to design a code of rate .85 from the ensemble $g(D, \rho)$ (In VHM systems codes with rates between .7 and .9 are typically used). Since it is very important to prevent the error floor (because a bit error rate of at least 10^{-12} is needed), we avoid degree-two variable nodes in the graph. We found the following degree distribution

$$d_1 = 3, d_2 = 4, d_3 = 7, d_4 = 10, d_c = 40. \tag{163}$$

Fig. 30 shows the performance of this code for the block lengths of 10000 and 100000 . As it is shown, at the bit error rate of 10^{-9}, the distances from the capacity are only .65 dB and 1.04 dB for the lengths 10^5 and 10^4, respectively. Moreover, the codes do not

present any error floor at BER of 10^{-9}. Another interesting property that is verified by our experiments is that it is almost always possible to find very good codes from the ensemble $g(D, \rho)$ having small maximum variable-node degree when the number of channels is greater than 3. This implies that we can avoid the high complexity degree distribution optimization. When the number of channels is 2 or 3, a small relaxation on the restrictions is needed. For example, one possibility is to allow two distinct degrees for the variable nodes of each type. In the next section we give some practical results on coding for VHM systems.

5.5.2 Application of Non-uniform LDPC Codes in Volume Holographic Memory Systems

In this section, our goal is to show that our non-uniform error correction scheme can significantly improve the storage capacity of volume holographic memory (VHM) systems. Holographic memories have been of intense interest recently due to their potentials for large storage capacity and fast data access. Recently, much research has been done on holographic storage systems, and several demonstrations of holographic memory systems have been reported [17, 50, 76, 108, 116] . The information in a holographic memory system is recorded and retrieved in the form of two-dimensional data pages, i.e., two-dimensional patterns of bits. During recording a page, a signal beam is formed by modulating a plane wave that is generated by a spatial light modulator (SLM). The interference of this signal beam with a reference beam is recorded in a recording medium. Several pages (at least 1000) are multiplexed in a holographic memory module using distinct reference beams for distinct data pages. Multiplexing of up to 10,000 holograms has been reported [5]. Read-out of a desired page is performed by the reference beam corresponding to that page. The diffraction of the reference beam off the hologram onto a camera (CCD or CMOS) results in the retrieval of the data page. The parallelism during recording and read-out due to the page-oriented nature of holographic memories results in large recording and read-out rates. The possibility of multiplexing several holograms in the same volume results in considerable data storage capacities. The recent advances in SLM and CCD technologies play a major role in the success of holographic memories as both the storage capacity and the data transfer rates scale linearly with the number of bits per page. Currently both SLM's and

CCD's with at least 1024 × 1024 pixels are available resulting in 1 Mbit pages. Multiplexing 1000 of such pages in a memory module (typical size of the recording material: 1 cm^3) results in a capacity of 1 Gbit. A modest frame rate of 1 kHz during read-out results in 1 Gbit/s data rate. With advances in both recording materials (which allows multiplexing more holograms) and the SLM and CCD technologies (which allow more pixels per page and larger frame rates), improvement by at least one order of magnitude in both the storage capacity and the data transfer rate is expected in the near future.

The capacity of a holographic memory system is controlled by the number of pages and the number of information bits per page. The number of pages (or holograms) is usually determined by the dynamic range of the recording materials. Multiplexing more holograms results in weaker holograms and lower signal to noise ratios. If M holograms are multiplexed appropriately, the diffraction efficiency (η) of each hologram is given by $\eta = [(M/\#)/M]^2$, with $M/\#$ being the dynamic range parameter [35]. Using weak holograms (corresponding to large number of pages) results in large raw bit error rate (typically $10^{-5} - 10^{-3}$). This is much higher than the practically required bit error rate (BER) of 10^{-12}. This makes the use of error correcting codes inevitable. The use of strong error correcting codes results in a smaller number of information bits per page due to larger number of parity bits added for error correction. On the other hand, since larger raw bit error rates are acceptable for stronger codes the number of pages is increased. Therefore, for a given error correcting code, there is an optimum number of holograms that results in the maximum storage capacity. This optimum depends on several parameters including the noise characteristics of the systems, the dynamic range parameter ($M/\#$), and the error correcting code. Read-Solomon codes and modulation codes have been extensively used for holographic memory systems [16, 22, 82]. The detailed optimization of the storage capacity of holographic memory systems using Reed-Solomon codes has been reported [22]. Soft decision array decoding and parallel detection for page-oriented optical memories have been also studied [21, 23].

The noise characteristics and therefore, BER in holographic memories is not uniform over a data page. Typically, the probability of error is minimum at the center of the page and increases by increasing the distance from the center of the page [22] (BER is highest at

the corners of the page). Typically, raw BER might vary by two orders of magnitude over a page.

Therefore, we need to design a nonuniform error protection scheme. Chou and Neifeld proposed an interleaving scheme to deal with the nonuniform error pattern arising from random and systematic errors [22]. They could increase the storage capacity by their interleaving method.

An excellent candidate for nonuniform error protection in holographic memory systems is the family of low-density parity-check (LDPC) codes. Our focus in this section is to show the potentials of LDPC codes for holographic memory systems. We will compare the performance of a typical storage system incorporating the LDPC codes with that incorporating the Reed-Solomon codes. We use a holographic system similar to that previously used for the optimization of the memory systems with the Reed-Solomon codes reported in [22]. We perform the optimization for the same system with the LDPC codes. Although we concentrate on the LDPC codes for holographic memory systems, the coding method presented here is general and can be applied to other page-oriented memory systems.Here, we use our methodology for LDPC over non-uniform channels.

Error correcting codes (ECC) have been applied to VHM in order to increase the storage capacity of the system. Storage capacity is defined as the number of information bits stored under the condition that BER is lower than a required value. The information theoretic capacity can be considered as an upper bound for the storage capacity. Since the diffraction efficiency of the recorded holograms decreases with increasing the number of pages, BER increases when we increase the number of stored pages. To increase the storage capacity, we can store more pages and use ECC to decrease BER to the desired value. If we increase the number of stored pages by a factor f, the capacity of the system is increased by the factor $f \times R$, where R is the code rate (the ratio of the number of information bits to the total number of bits) . Thus, for a constant number of pages , in order to have the highest storage capacity we need to find a code with highest rate that provides us with the required output BER. The optimization of the number of pages was studied in [22] . Here, we first assume a fix number of pages and try to design codes for VHM with R as large as possible

while keeping the BER constant. Then we change the number of pages and try to maximize the storage capacity.

The decoder of an ECC can be a soft-decision decoder or a hard-decision decoder. In the hard-decision decoding, inputs to the decoder are binary-valued bits. Unlike the hard-decision decoding, the inputs in the soft-decision decoding are real numbers (in practice an analog-to-digital converter is used to quantize the input to a finite number of levels). Consider a VHM system in which we assume all pixels are independent. Note that in reality, pixels are not independent; however, we make this assumption to make our analysis easier. The information theoretic capacity of this system is equal to

$$C = M \sum_{i=1}^{N^2} C_i \tag{164}$$

where M is the number of stored pages, N^2 is the number of pixels in a page and C_i is the capacity of the channel seen by the i^{th} pixel. Note that C_i depends on M. If we have access to only hard information of the output of the channel, then the channel can be considered as N^2 parallel binary symmetric channels (BSC). The information theoretic capacity of this channel model is

$$C = M \sum_{i=1}^{N^2} C_i = M \sum_{i=1}^{N^2} (1 - H(p_i)), \tag{165}$$

where p_i is the probability of error of the i^{th} bit and H is the binary entropy function given by

$$H(p) = p \log_2(\frac{1}{p}) + (1-p) \log_2(\frac{1}{1-p}). \tag{166}$$

However, if we have access to the soft information in the decoder and if we assume the additive white Gaussian noise approximation, then the channel can be modelled as N^2 parallel binary input additive white gaussian noise (BIAWGN) channels for which the capacity C_i is given by [112]

$$C_i = -\int \phi_i(x) \log_2(\phi_i(x)) dx - \frac{1}{2} \log_2(2\pi e \sigma_i^2), \tag{167}$$

Here, we have

$$\phi_i(x) = \frac{1}{\sqrt{8\pi\sigma_i^2}}(e^{-\frac{(x+1)^2}{2\sigma_i^2}} + e^{-\frac{(x-1)^2}{2\sigma_i^2}}) \tag{168}$$

where σ_i is the variance of the noise that affects the i^{th} bit. Figure 31 depicts the capacity of the BIAWGN and BSC channels versus the bit error probability. Obviously, the capacity of the BIAWGN channel is higher than that of the BSC with the same bit error probability because in BIAWGN channel we have more information about the output of the channel. There exist both soft and hard-decision decoding algorithms for LDPC codes [38, 112]. To have the best BER performance, we choose to perform soft-decision decoding as we explain later.

LDPC codes are suitable for holographic memories for a variety of reasons . First, it is shown that they have a performance near Shannon limit [24, 54, 112]. Therefore, we will be able to approach the information theoretic capacity of the channel using LDPC codes, while RS codes do not have a performance near the information theoretic capacity for the practically limited block length. Second, not only do we use the prior knowledge of the noise distribution in the VHM data page in designing the code, but also we use this information in the decoding period. On the contrary, it is not easy (if not impossible) to incorporate the prior knowledge of noise distribution into the designing and decoding of RS codes. An interesting method was proposed by [22] to cope with the nonuniform noise distribution. The authors suggested to interleave the bits such that all message blocks contain the same number of good bits and bad bits(bits with low noise and bits with high noise). In other words, the average noise power in a message block after interleaving is independent of the location of bits. However, we still cannot use the prior information about noise distribution at the decoding step.

In the design of LDPC codes we use the flexibility of these codes for choosing the degree distribution of the Tanner graph. We choose the degree distribution such that the code performance is optimized for the channel noise distribution. In decoding process, we use log-likelihood ratios (LLR) that contain the information about the noise power for a specific bit and the information about how reliably that bit was transmitted across the channel.

Third, the decoding of LDPC codes is fully parallelizable and very fast, which makes these codes desirable for VHM systems. This allows us to use a long block length and decrease BER while we maintain lower redundancy.

The main drawback of LDPC codes is that they have a slow encoder. This is not a problem in the VHM systems because we use a high rate LDPC code with systematic encoder. Therefore, we need to encode only parity bits whose number is a small fraction of the block length. Moreover, we can also use the method described in [113] to simplify the encoding process.

Another problem with LDPC codes is that they may show an error floor effect. However, not all LDPC codes have this property. For example, for the LDPC codes that we designed in this section, we did not observe any error floor down to the BER of 10^{-9}. Additionally these codes perform close to the Shannon capacity. An alternative technique to deal with error floor is to concatenate an outer code with an LDPC code. This way, we can decrease error probability significantly , with a small loss of the storage capacity. However, an interleaver is required to distribute the errors in an erroneous LDPC word to several words of the outer code.

We mention that when we change the number of pages, we need to design a new LDPC code with a different degree distribution so that the code is optimized for the new channel. However, this is not a problem because the code is designed off-line. Moreover, this flexibility of LDPC codes allows us to optimize the code for each specific channel. On the contrary, for the RS codes over $GF(q)$ ($GF(q)$ is the finite field with q elements), there is no need for designing because there is no design parameter, except the rate.

As we mentioned, since the error pattern in a page is non-uniform, we use the ensemble $g(\Lambda, \rho)$ for VHM systems. Generally, pixels at the corner of a data page have higher probability of error than those at the center of the page. We now undertake some important issues about the design of these codes. Let us consider the encoding problem. Since an LDPC code is used with very large lengths, its generator matrix has large dimensions. This requires a large number of computations in the encoding algorithm. To avoid this, we use the generator matrix G in the systematic form. This means that if we encode a vector

$(u_1, u_2, ..., u_k)$ to a codeword $(x_1, x_2, ..., x_n)$ we have $u_i = x_i$ for $1 \leq i \leq k$. Therefore, we need to calculate only $n - k$ bits $x_{k+1}, x_{k+2}, ..., x_n$. Since in holographic memories, we usually use high-rate codes, $n - k$ is a small number that results in less computation with respect to nonsystematic encoding.

Another issue is avoiding short cycles in the Tanner graph of the code. In order to have a good performance, we need to avoid short cycles (cycles of length four) in the Tanner graph [38, 72]. Unfortunately, the higher the code rate is, the more difficult (if not impossible) is to eliminate these cycles. Since we use high- rate codes in holographic memories, it is likely that there exists lots of short cycles in the Tanner graph representation of the code. We omitted these short cycles as much as possible. We also maintained the graph very sparse (by choosing a very sparse parity check matrix) to avoid the short cycles.

We would like to point out that in [23], authors proposed a likelihood-based two-dimensional equalization for extenuating interpixel interference (IPI) noise in VHM systems and combined it with the soft-decision of the array codes. A similar scheme can be used for LDPC codes as well to improve the performance of the code further. The decoding algorithm for LDPC codes in intersymbol interference (ISI)channels is described in [59].

5.5.3 Simulation Results for VHM Systems

We implemented the LDPC codes that we designed to examine their performance. For simulation we chose a system similar to [22] . As explained in [22], different kinds of errors are present in the system. The probability density function (pdf) of the noise is determined by considering the effect of all these error sources. For simplicity, we assume the noise is additive white Gaussian and its variance is a function of the pixel location. Note that the formulas used in decoding and density evolution are quite general and can be applied to any symmetric [112]noise distribution. Therefore, our analysis can be applied to any system with a nonuniform error pattern. As mentioned before, the raw bit error rate in volume holographic storage depends on the position of the bit in the data page. Figure 32 shows the different regions with constant raw BER before error correction. In each region, pixels have almost the same probability of error [22]. In our simulations we divided a page into four

regions. We assume the system has a raw bit error rate roughly from 10^{-3} to 10^{-6} when 2000 pages are stored. This raw bit error rate increases when the number of pages increases. Similar to [22] we make the following assumption: The magnitudes of the systematic error and the thermal noise are assumed to remain unchanged with respect to M(the number of pages) and SNR per pixel can be computed by using the scaling law which states that the SNR is proportional to $\frac{1}{M^2}$ [14, 22].

Normally, the output BER of 10^{-12} is desirable for the holographic storage. However, because of the extensive computation that is involved to find the performance of the code at 10^{-12}, we are imposed to obtain an upper bound on the BER. Since it is computationally feasible to decode 10^9 bits, we performed our experiments for this number of bits. For an optimized LDPC code of a given rate we found the maximum number of pages such that after the decoding of 10^9 bits, no error was observed. We then concluded that the average BER was upper bounded by 10^{-8}. We also considered an RS codes of several different lengths ranging from 15 to 511 and determined the number of pages for the output BER of 10^{-8} . We expect that if the actual error rate for the LDPC code is higher than 10^{-12}, we can reach the BER of 10^{-12} by very subtle reduction in the capacity provided we do not face an error floor problem. The reason for this is that LDPC codes are known to have a threshold effect [112]. For a given degree distribution, this threshold can be defined as the maximum possible noise level, in order to have reliable communication. Equivalently, we can define the SNR threshold as the minimum SNR required for reliable communication. If the SNR is higher than the SNR threshold we can achieve arbitrary small probability of error if we are allowed to have a high enough block length. However, if SNR is lower than the SNR threshold, the probability of error is bounded away from zero by a strictly positive constant. As long as we use these codes for a channel with SNR higher than the threshold, increasing SNR by a small value, results in a drastic reduction of the BER [24]. Since we use these codes just bellow their noise threshold (or above the SNR threshold), we expect that even if our codes have a BER higher than 10^{-12}, we can reach this error rate by reducing the number of pages slightly. The above discussion is valid if the code does not have an error floor higher than 10^{-12} . In case that we cannot avoid the error floor, as

we mentioned before we can concatenate an outer code with the LDPC code.

Figure 33 shows the storage capacity that is obtained by using LDPC codes and RS codes of different lengths and different decoding methods. For RS codes, we used the same interleaving scheme that was proposed in [22] to improve the performance of the code for the nonuniform noise distribution. Only hard decision decoding is considered for RS codes. The maximum storage capacity that is gained by using RS codes is .5609 Gbits which is obtained when an RS code of length 511 is used and 2802 pages are stored. The maximum storage capacity that is obtained by using LDPC codes is .8423 Gbits. This is achieved when 4600 pages are stored and the soft LDPC decoder is used. We note that this capacity is about 50 percent higher than that of the RS codes. This sizable increase in the capacity by the LDPC code can be explained by using Figure 33. When the number of pages is small, there is not much difference between the RS codes and the LDPC codes. This is because the information theoretic capacity of hard-decision and soft-decision decoding are close to each other for high SNRs (i.e., small number of pages (Figure 31)). Moreover, RS codes have a good performance for such SNR's. However, when the number of pages increases and therefore SNR decreases, the difference between the capacity of hard-decision and soft-decision decoding increases. More importantly, LDPC codes maintain near the Shannon limit performance for the low SNR while the performance of RS codes is far from Shannon limit in the low SNR. For this reason, the optimum number of pages for LDPC codes is higher than that for RS codes. We also note that the performance of LDPC codes with hard decision decoding is about 25 percent higher than the maximum capacity of the RS codes.

It is important to note that the full advantage of LDPC codes is obtained if we choose the optimum number of holograms ($M \simeq 4600$). The number of holograms that can be recorded in a recording material (for example, a photo refractive crystal) is limited by the finite dynamic range and the angular selectivity. Using a 1cm thick $LiNbo_3$ crystal with the current values of $M/\#$, it is possible to multiplex several thousand holograms. Two reported examples are 5000 and $10,000$ holograms [5, 79]. If for any reason (thin crystal, small $M/\#$, large noise level, etc.) the maximum number of holograms is bellow 2000, the

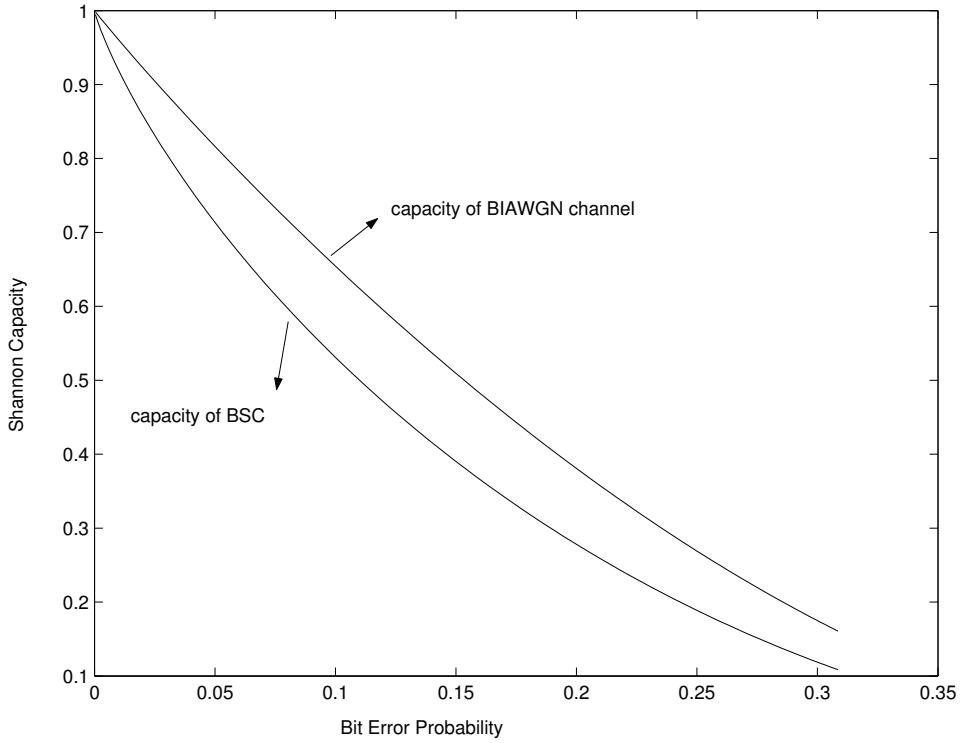

Figure 31: Capacity of BSC and BIWAGN channel versus bit error probability

advantage of LDPC codes will be lost as evidenced by Figure 33.

5.5.4 Simulation Results for Unequal Error Protection

In this section we describe experiments to measure the performance of some codes from the ensemble $g_{n'}(\lambda, \rho, h, d_v, d_c)$. We showed in previous sections that the asymptotic performance of the UELDPC codes is good. Thus, here we concentrate on finite-length UELDPC codes and show that these codes have good performance even for short block lengths. To design UELDPC codes we first need to choose h. Note that taking $h = 0$ results in an ordinary LDPC code from the ensemble $g_n(\lambda, \rho)$ in which we assign the high-degree variable nodes to the important bits. As we mentioned before, this is not an efficient approach. Our experiment shows that usually $h = 2$ results in good codes. For short-length codes (with lengths between 1000 and 5000), it is not suitable to choose h larger. We first consider the binary erasure channel. We designed half-rate UELDPC codes of lengths $n = 2000$ and $n = 4000$. The value of $\xi(n)$ was chosen 50 and 100 for $n = 2000$ and $n = 4000$ respectively. Let c' be either one of these codes and $g_{c'}$ be its corresponding graph. Let A be the set of

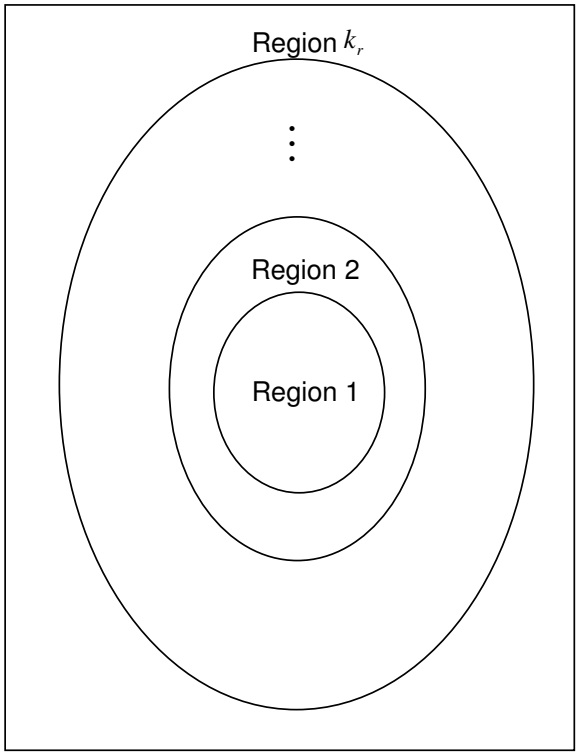

Figure 32: Different regions in a typical data page in holographic recording. Raw BER is almost constant in each region.

important variable nodes. We chose the degree of the important bits as $d_v = 12$ and the degree of the vertices in $N(A)$ as $d_c = 5$. The degree of the vertices in $N^2(A)$ was one of the values 8, 3, and 2. The degrees of all the other check nodes was 8. The degrees of the rest of the variable nodes were either two or three. To construct UELDPC codes we use a method similar to one described in [112]. We assign sockets to the vertices and construct the graph using a random permutation. The only difference with [112] is that the permutation that we use is a restricted random permutation to make sure that the vertices in $N^0(A) = A$, $N(A)$, $N^2(A)$, ..., $N^h(A)$ take the desired degrees.

Fig. 34 shows the performance of the code c' when the length of the code is $n = 2000$. It also shows the performance of the regular $(3, 6)$ code of length 2000 which is the best regular LDPC code. We did not consider irregular codes because their performance is only slightly better than the regular codes for short lengths. Moreover, there is no efficient method to find good short-length irregular codes. In Figure 34, by bad bits we mean the bits other than the important bits in the code c'. The figure shows both important bits and

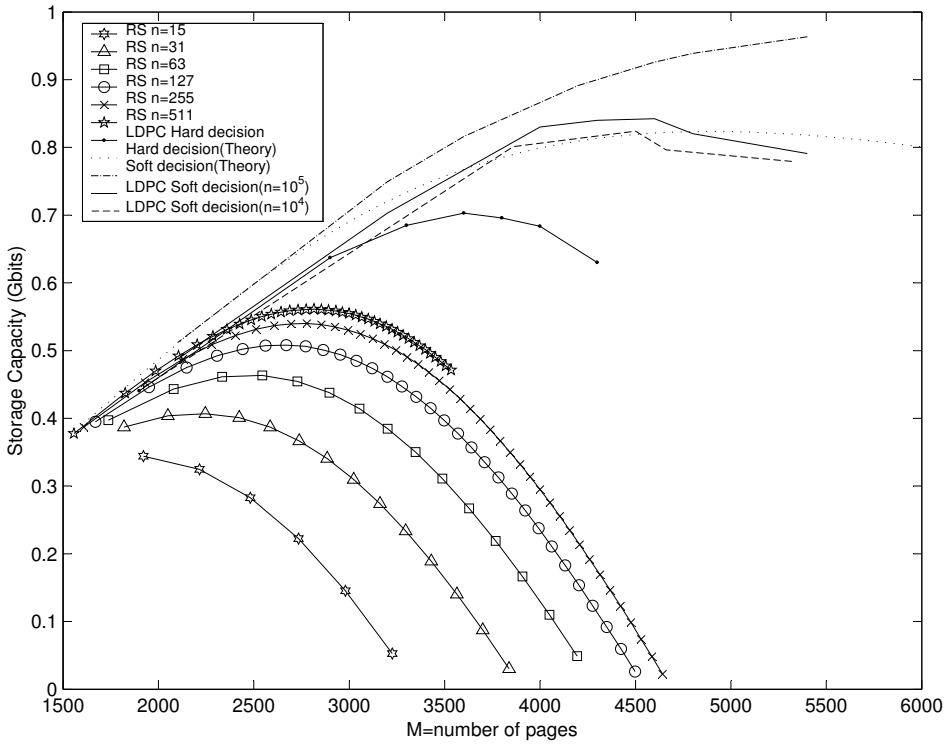

Figure 33: comparison of different coding schemes for VHM's.

the bad bits have smaller bit error rates than the bit error rate of the $(3, 6)$ regular code. In particular, the important bits have a much lower bit error rate. Fig. 35 shows the same results when the lengths of the codes are $n = 4000$. We observe that the results are similar to the case of $n = 2000$.

Let us now consider the binary-input additive white Gaussian noise (BIAWGN) channel. Again let us consider the code c' and compare its performance with the $(3, 6)$ regular code. Although we designed this code for the binary erasure channel, it is useful to evaluate its performance over the BIAWGN channel. Fig. 36 shows the performance of the code c' and the regular code for the length $n = 2000$. We notice that both bad bits and important bits of the code c' have better bit error rates than the regular code. Additionally, the bit error rate of the important bits is considerably lower than the bit error rate of the regular code. Since the $(3, 6)$ regular code is considered to be a good code for length $n = 2000$, we conclude that the code c' is also a good UELDPC code for the BIAWGN channel.

At the end we would like to emphasize that the assumption $\xi(n) << n$ is crucial for the

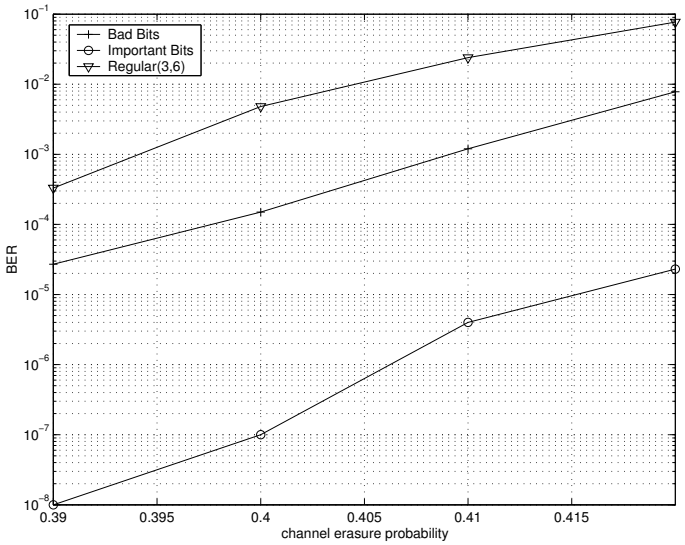

Figure 34: Comparison between the performance of an UELDPC code of length 2000 and the regular (3,6) code of the same length over the binary erasure channel.

above proposed scheme. For finite-length cases, if the ratio of the most important bits is comparable with the code length, it is still possible to design UELDPC codes by choosing higher degrees for these bits. However, it is an open question whether this code would be efficient compared with two separate LDPC codes.

5.6 Other applications

The proposed framework to design LDPC codes for the non-uniform error correction has several other applications. Multicarrier orthogonal frequency division multiplexing (OFDM) and multilevel coding are among these applications. The OFDM systems consist of several parallel channels in which some bits experience higher SNR's than others. Thus we need to perform non-uniform error protection. In this case, we are not usually concerned about the error floor problem. Instead we have other restrictions. Specifically, we may not be able to use long codes. It can be shown by simulations that for finite-length cases, the codes from the ensemble $g(\Lambda, \rho)$ may perform better than the conventional irregular codes over the parallel channels. A similar situation exists in multilevel coding.

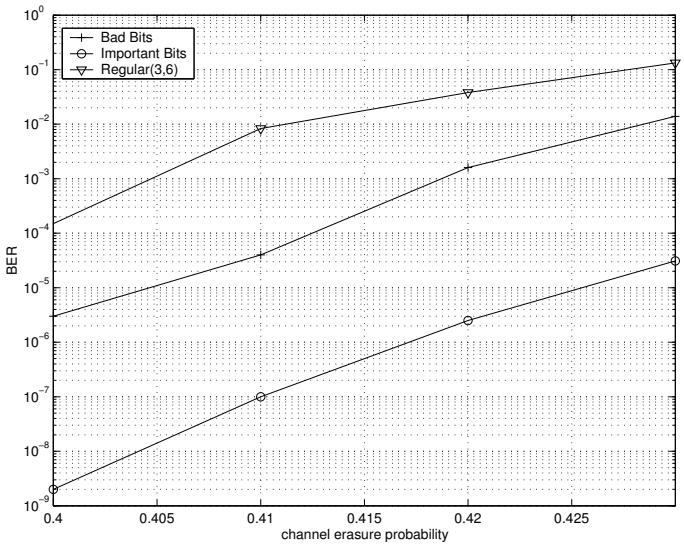

Figure 35: Comparison between the performance of an UELDPC code of length 4000 and the regular (3,6) code of the same length over the binary erasure channel.

5.7 Conclusion

In this chapter we proposed a framework to design good LDPC codes over a set of parallel channels. This method is useful for many applications such as the volume holographic memories, OFDM, and rate-adaptive coding systems. We showed that the proposed method has several advantages over the conventional method. First, the design procedure is very simple since we do not need to perform the high-complexity degree optimization algorithms that are necessary for conventional LDPC codes. Second, using the proposed method, we can find codes that have near Shannon-limit performance and have lower error floor. Third, for the applications that the code length cannot be large, the proposed codes can have better performance than the ordinary LDPC codes. We used our non-uniform coding schemes in VHM systems and showed that we can increase the storage capacity by more than fifty percent compared to previous schemes. The proposed framework can also be used to design LDPC codes in other applications such as OFDM systems and multilevel coding.

We also showed that the analysis and optimization of rate-compatible LDPC codes can be done as a special case of our analysis for parallel channels. The developed LDPC code employs single encoder and decoder for all combination of rates that are desired. As opposed to traditional rate-adaptive convolutional codes, we can generate any combination of rates

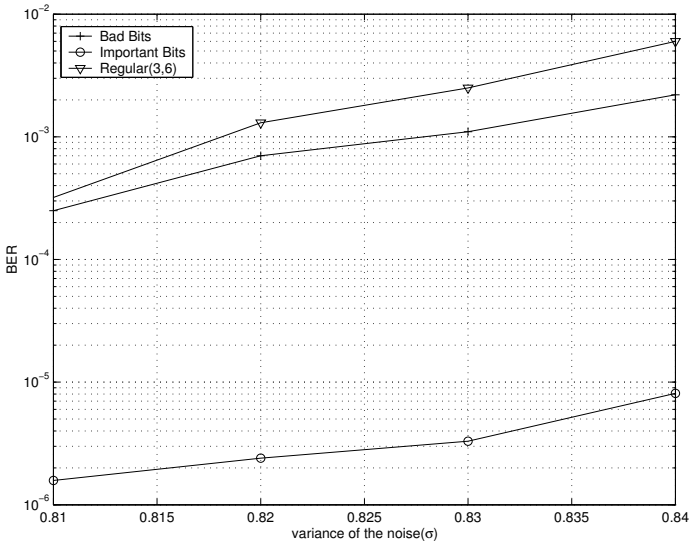

Figure 36: Comparison between the performance of an UELDPC code of length 2000 and the regular (3,6) code of the same length over the BIAWGN channel.

very easily.

Finally, we investigated unequal error protection using LDPC codes. In particular, we showed that good UELDPC codes exist for certain applications in which a small fraction of bits should be strongly protected. We proposed a technique to design these codes. We showed that these codes are asymptotically as good as any equal protecting LDPC codes. For short-length codes, simulations demonstrate that these codes outperform regular LDPC codes. Additionally, with the proposed scheme we can decode the important bits without having to decode the entire block. In ongoing research, we are exploring this issue further.

CHAPTER VI

RATE-COMPATIBLE CODES

6.1 Introduction

In this chapter we study two different schemes for obtaining rate-compatible codes: punctured LDPC codes and Raptor Codes. We study punctured LDPC codes and show some fundamental properties of these codes [100,103]. In chapter 5, we studied punctured LDPC codes in the context of coding for non-uniform channels. Here we study punctured LDPC codes in more detail. We prove that punctured LDPC codes have a threshold effect and compute the threshold for different methods of puncturing. We specifically show that arbitrary rates are achievable via puncturing. We then discuss the optimality of punctured LDPC codes. For the BEC, much stronger results are obtained. For example, using only one encoder and decoder, we can achieve the capacity of BEC on arbitrary set of rates. We discuss design of good puncturing schemes for LDPC codes and we propose a simple rule for constructing rate-compatible LDPC codes. The proposed method prevents the performance degradation for the high rates that was previously observed in [46]. It is also applicable to finite-length LDPC codes. Finally we consider the open research problem of capacity achieving sequences for general memoryless binary-input output-symmetric (MBIOS) channels. We prove that if capacity achieving sequences of LDPC codes exist when the rate of the codes approaches zero, then capacity achieving LDPC codes exist for all rates.

We then consider Raptor Codes. We introduce a construction of Raptor codes to for symmetric channels. An important property of the proposed scheme is that unlike the previous construction of Raptor Codes, it allows to design codes that are approaching the capacity of the underlying channel in a rate-compatible way.

Throughout the chapter we assume the following terminology. By a graph we mean a simple graph, i.e., a graph with no loops (edges joining a vertex to itself) and no multiple edges (several edges joining the same two vertices). Let A be a subset of the vertices in the

graph g. $N(A) = N^1(A)$ shows the set of neighbors of A in g. More generally, for $j \in \mathbb{N}$, $N^j(A)$ is the set of vertices in g from which there is path of length j to a vertex in A. Let D be a subgraph of g such that its vertex set is A. We say D is induced by A if D contains all edges of g that join two vertices in A. For a graph g, $deg_g(v)$ is the degree of v in g. If V is the set of vertices in g and $U \subseteq V$, then $deg_U(v)$ is the number of neighbors of v in U. For a random variable X, we show its distribution by $F_X(x)$. If the random variable has a well-defined density function, we represent the density function by $f_X(x)$. Similar to [54], we define $P_e(F_X) = \Pr\{X < 0\} + \frac{1}{2}\Pr\{X = 0\}$.

6.2 Punctured LDPC codes

In this section, we study some fundamental properties of punctured LDPC codes. Puncturing is one of the most common methods to construct rate-compatible codes. In this method, to change the rate of a code to a higher rate, we puncture (delete) a subset of the codeword bits. We first study the puncturing capacity of LDPC code ensembles. Particularly, we prove that any LDPC code ensemble has a puncturing threshold p^*. If the puncturing fraction p is smaller than p^*, then the punctured code is asymptotically good. In other words, a code from the ensemble can be used to achieve arbitrarily small error probability over a noisy channel while the code rate is bounded away from zero. On the other hand, if $p > p^*$, error probability is bounded away from zero, independent of the communication channel. The puncturing thresholds can be easily computed for both randomly and intentionally punctured LDPC codes. We also show that puncturing is a lossless process in the sense that for any ensemble of LDPC codes of rate R_1, there exists a punctured LDPC code ensemble of an arbitrary rate $R_2 < R_1$ with the same threshold under the message passing decoding algorithms. We then show that for any rates R_1 and R_2 satisfying $0 < R_1 < R_2 < 1$, there exists an LDPC code that can be punctured from rate R_1 to R_2 such that the resulting code is asymptotically good for all rates R, $R_1 \leq R \leq R_2$. Specifically, this shows that rates arbitrarily close to one are achievable using puncturing. For BEC, we show that using only one encoder and decoder and a proper puncturing scheme, we can achieve the capacity for an arbitrary set of positive rates. We then propose a method to design good punctured

LDPC codes over a broad range of rates. The method is very simple and does not need any optimization process. Additionally, it avoids performance degradation at high rates. It is also applicable to finite-length LDPC codes. Finally, we show the possible application of punctured LDPC codes for proving the existence of capacity achieving sequences for MBIOS channels.

We consider the following scheme. We take an LDPC code of rate $R_p = k/n$, where k and n are the length of the information blocks and the codewords, respectively. To generate a code with a new rate, we puncture a subset of bits in the codeword and send the unpunctured bits to the receiver. It is assumed that the decoder knows the position of punctured bits in the codeword. To start the decoding, we need to compute log likelihood ratios (LLR's) in the decoder. The LLR's for the punctured bits are zero.

Ha and McLaughlin studied optimal puncturing of LDPC codes [46]. They studied two methods for puncturing LDPC codes. In the first method, the authors chose the punctured bits randomly (i.e., if the puncturing fraction is p, they chose a subset of the bits in the codeword with cardinality pn at random and puncture the bits in the subset). This method is called random puncturing. In the second method, the intentional puncturing, they optimized the puncturing distribution. The authors set up the intentional puncturing as follows. First, variable nodes of the bipartite graph were grouped in accordance with their degrees. Then, they randomly punctured a fraction π_j of the nodes in G_j, where G_j is the set of variable nodes of degree j.

As we mentioned, a punctured code can be modelled as a code that is used over two parallel channels as Figure 14. In this model, punctured bits are transmitted over the second channel that has a zero capacity. Let us define

$$\phi_j = \frac{\lambda'_j \pi_j}{\Sigma_j \lambda'_j \pi_j}, \quad p = \Sigma_j \lambda'_j \pi_j \tag{169}$$

where λ'_j is the fraction of degree-j variable nodes in the graph. Thus if we let $\Phi = \{\phi_j : j = 2, 3, ..., d_{v_{max}}\}$, an intentional puncturing distribution for a code from (λ, ρ) ensemble can be represented by the pair (Φ, p) in which p shows the puncturing fraction and Φ determines the puncturing structure. A valid puncturing pattern is obtained when we have $p\phi_j \leq \lambda'_j$,

for $j = 2, 3, ..., d_{v_{max}}$. It is worth noting that asymptotically random puncturing is a special case of intentional puncturing, as stated in the following fact.

Fact 1. *A randomly punctured ensemble of LDPC codes with a puncturing fraction p has the same threshold as the intentionally punctured code with puncturing distribution* $\phi_j = \lambda'_j$, *for* $j = 2, 3, ..., d_{v_{max}}$.

Proof. The proof is simple and can be done by evaluating the density evolution formulas for the punctured codes given in [106]. The proof alternatively follows from Strong Law of Large Numbers (as $n \to \infty$, both of the distributions become the same almost surely), and using continuity of the density evolution over space of distributions. □

Fact 1 implies that any asymptotic result that is valid for intentionally punctured codes is usually valid for randomly punctured codes. Thus, whenever we are concerned with asymptotic properties, we only give the result for intentional puncturing. As we mentioned in [106] a punctured LDPC code can be modelled as Figure 37. In this figure the punctured bits are transmitted through a channel with zero capacity. The actual channel, is a MBIOS channel with the parameter θ. When θ decreases, i.e., when the channel improves, we increase the puncturing fraction. Thus we can consider the puncturing process as a change in the channel not in the code rate.

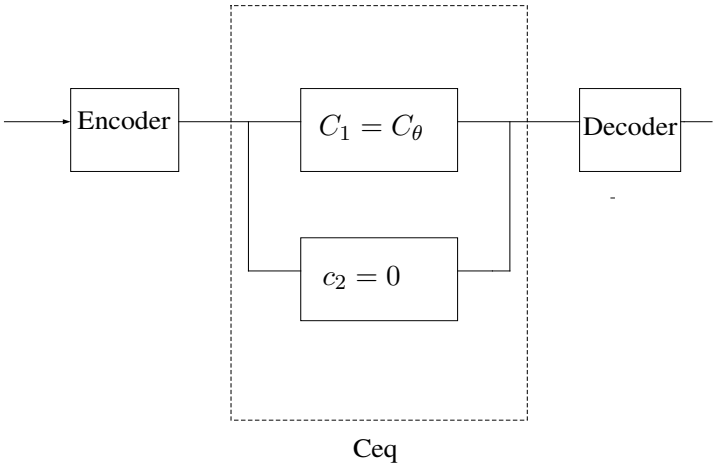

Figure 37: A model that describes puncturing over a binary channel

6.2.1 Puncturing threshold of LDPC codes

In this section we compute the puncturing threshold p^* of LDPC code ensembles. Again consider our terminology for physically degraded MBIOS channels. That is consider an MBIOS channel with parameter θ, where $\theta \in [\theta_{min}, \theta_{max}]$ and $\theta_{min}, \theta_{max} \in \mathbb{R} \cup \{-\infty, +\infty\}$. Remember $\mathcal{C}$ shows a class of channels with parameter θ, and a channel in $\mathcal{C}$ with parameter θ_0 is called C_{θ_0}. The capacity of the channel C_{θ_0} is shown by c_{θ_0}. For simplicity, we assume that c_θ is a continuous function of θ, and if $\theta_1 < \theta_2$, then C_{θ_2} is physically degraded with respect to C_{θ_1}. For the channel C_{θ_0} assuming the all-one code word has been sent, we define the random variable Z_{θ_0} as

$$Z_{\theta_0} = \ln \frac{p(X=1|Y=y, \theta=\theta_0)}{p(X=-1|Y=y, \theta=\theta_0)}, \tag{170}$$

where X and Y are the input and output of the channel, respectively. Assume that if $\theta_1 < \theta_2$, then C_{θ_2} is physically degraded with respect to C_{θ_1}. For simplicity, we assume that $P_e(F_{Z_\theta})$ is a continuous function of θ and we have $\lim_{\theta \to \theta_{min}} P_e(F_{Z_\theta}) = 0$. Equivalently we can say that as θ tends to θ_{min}, Z_θ tends to infinity in probability [54] and thus the probability density function of Z_θ, f_{Z_θ}, converges to Δ_∞ [112], [54]. Furthermore, we may assume that if $\theta > \theta_{min}$, then $P_e(F_{Z_\theta}) > 0$. Note that almost all practical MBIOS channels such as BIAWGN channel, BSC, and BEC satisfy these properties.

We say an ensemble of LDPC codes of positive rate R is asymptotically good if there exists a $\theta_1 \in [\theta_{min}, \theta_{max}]$ such that $P_e(F_{Z_{\theta_1}}) > 0$ and a randomly chosen code from the ensemble can be used to achieve arbitrary small error rate over C_θ for all $\theta \leq \theta_1$. On the other hand, if we need $P_e(F_{Z_{\theta_1}}) \longrightarrow 0$ for achieving arbitrarily small error rate, the ensemble is said to be asymptotically bad. For a punctured ensemble (λ, ρ, Φ, p) we define

$$\lambda_i^{(2)} = \frac{\lambda'_i i \pi_i}{\Sigma \lambda'_j j \pi_j}, \quad q = \frac{\sum j \pi_j \lambda'_j}{\Sigma j \lambda'_j}. \tag{171}$$

We also define the following sequence.

$$x_0 = 1, \quad x_l = \lambda^{(2)}\big(1 - \rho(1 - q x_{l-1})\big). \tag{172}$$

Then, we have the following theorem [100].

Theorem 21. *(Puncturing threshold of LDPC codes) Consider the ensemble of LDPC codes defined by the pair* (λ, ρ) *that are intentionally punctured by the puncturing pattern* (Φ, p)*. Assume a randomly chosen code from the punctured ensemble is used over the channel* C_θ*, with* $\theta > \theta_{min}$*. Let* p_{th} *be supremum value of the puncturing fraction* p *such that in (172), we have*

$$\lim_{l \to \infty} x_l = 0. \tag{173}$$

If $p > p_{th}$*, then the decoding error probability is bounded away from zero independent of the communication channel. On the other hand if* $p < p_{th}$*, then there exists a* $\theta^* > \theta_{min}$ *such that if the channel parameter* θ *is smaller than* θ^**, then*

$$\lim_{l \to \infty} P_e(F_l) = 0. \tag{174}$$

The above result shows that p_{th} is the threshold of punctured LDPC codes. Using Theorem 21, we can find the puncturing threshold of randomly and intentionally punctured LDPC code ensembles. The cut off rate of the code R_{th} is obtained by

$$R_{th} = \frac{R_p}{1 - p_{th}}, \tag{175}$$

where R_P is the rate of the parent code. For example, since the $(3, 6)$ regular ensemble has puncturing threshold (for regular codes the random and intentional puncturing are the same) $p_{th} = .4294$, it has the cut off rate $R_{th} = .8763$. Thus, using the $(3, 6)$ regular ensemble as the parent code ensemble, we should not try to obtain any rates higher than .8763.

6.2.2 Achieving Arbitrary Rates Via Puncturing

In [46], authors evaluated the performance of several punctured LDPC codes and optimized the puncturing pattern to get the best performance. However, their simulations show that the performance of LDPC codes, degrades for high rates. Thus, we need to pay a big penalty for using punctured codes. This phenomenon can be explained by the threshold effect of punctured codes discussed in the previous section.

In [46], authors used three ensembles of LDPC codes for puncturing. These ensembles are

$$\lambda_2(x) = 0.25105x + .30938x^2 + .00104x^3 + .4385x^9,$$

$$\rho_2(x) = .63676x^6 + .36324x^7,$$

$$\lambda_3(x) = 0.23403x + .21242x^2 + .14690x^5 + .10284x^6 + .30381x^{19},$$

$$\rho_3(x) = .71875x^7 + .28125x^8,$$

$$\lambda_4(x) = 0.267817x + .204657x^2 + .077459x^5 + .2041810x^7 + .2458860x^{29},$$

$$\rho_4(x) = x^5.$$

The cut off rates of these ensembles are shown in Table 6.

Table 6: The Cut off rates of some LDPC code ensembles (for random puncturing).

Ensemble	Cut Off Rates
(λ_2, ρ_2)	.944
(λ_3, ρ_3)	.9301
(λ_4, ρ_4)	.9463

From Table 6, we can easily explain the degradation in the performance of the above ensembles at high rates. By examining the simulation results in [46], we observe that all of the above ensembles show considerable degradation for the rates above .85. The degradation seems to have a very high slope for the rates above .9. This is because at these rates we are approaching the cut off rate.

Using the results of the previous section, by suitably choosing the ensemble, rates arbitrarily close to one can be obtained via puncturing. In fact, this is achieved by using good codes for the BEC as the parent code and randomly puncturing them (off course intentional puncturing works, as well). Since capacity achieving sequences of LDPC codes over the BEC are known [70], [118], [119] and [84], we can find codes with cut off rates arbitrarily close to one. Moreover, the parent code can have any rate. Thus we have the following theorem [100].

Theorem 22. *For any rates R_1 and R_2 that $0 < R_1 < R_2 < 1$, there exists an ensemble of LDPC codes with the following property. The ensemble can be punctured from rate R_1 to R_2 resulting in asymptotically good codes for all rates $R_1 \leq R \leq R_2$.*

Theorem 22 ensures that we can have punctured LDPC codes that are asymptotically good (in the sense defined here) on an arbitrary set of rates. However, it does not give any clue a bout the gap from the capacity. Here we give a lower bound on the achievable rates as the channel parameter changes.

Theorem 23. *For any $\delta > 0$ and MBIOS channel C_θ parameterized by $\theta \in [\theta_{min}, \theta_{max}]$, and $R_1, R_2 \in (0,1)$, $R_1 < R_2$, there exists an ensemble (λ, ρ) of LDPC codes with the following properties. The ensemble can be punctured in a rate-compatible way to produce all rates $R \in [R_1, R_2]$. Moreover, for all $R \in [R_1, R_2]$ it can be used to reliably transmit data over C_{θ_R} satisfying*

$$R > 1 - E[e^{(-Z_{\theta_R}/2)}] - \delta, \tag{176}$$

where Z_{θ_R} is defined in (170).

Note that for any fixed rate, we can conclude the existence of an ordinary LDPC code ensemble that satisfies the above lower bound from [60]; however, the importance of Theorem 23 is the fact that we can have only one code that is simultaneously satisfies the bound for all rates.

Proof. Choose an LDPC code of rate R_1 whose threshold over the BEC, ϵ_{th} , satisfies

$$R_1 > 1 - \epsilon_{th} - \frac{R_1}{R_2}\delta. \tag{177}$$

We use this code as parent code and use random puncturing to obtain all rates R_p, $R_1 \leq R_p \leq R_2$. Now if $R_1 \leq R_p \leq R_2$, using Fig. 37, we can assume we still have a code of rate R_1, because as it was mentioned previously, puncturing can be considered as a change in the channel instead of the code rate. Now if we apply Theorem 4.2 of [60] to this system, we conclude the following. If

$$p + (1-p)E[e^{(-Z_{\theta_{R_p}}/2)}] = \epsilon_{th}, \tag{178}$$

then the threshold of the punctured code, θ_{th} satisfies $\theta_{th} \geq \theta_{R_p}$. Thus the punctured code can be used for reliable communication over $C_{\theta_{R_p}}$. However, using [177], [178], and $R_1 \leq R_p = \frac{R_1}{1-p} \leq R_2$, we conclude

$$R_p > 1 - E[e^{(-Z_{\theta_{R_p}}/2)}] - \delta. \tag{179}$$

□

Figure 38 shows the ratio of the achievable rate and the channel capacity for BSC. Figure 39 shows the gap from the capacity for the BIAWGN channel. As we see the gap is always less than 1.6 dB. The bound of Theorem 23 is interesting because it gives an analytical result; however, in practice we can find punctured LDPC codes that have better performance.

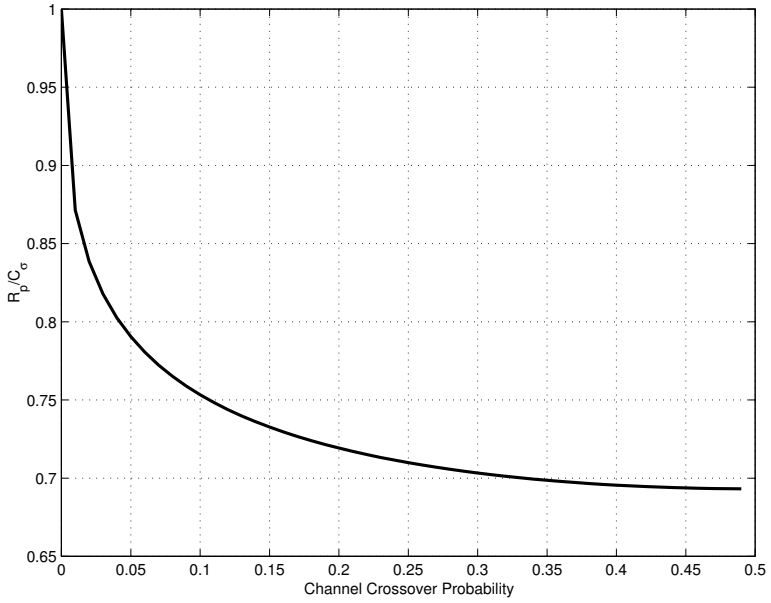

Figure 38: The ratio of the achievable rate and the capacity for an ensemble of punctured LDPC codes over BSC.

6.2.3 Optimality of Punctured LDPC Codes

In this section we show that by using punctured codes we do not lose performance. In other words, we show that for any LDPC ensemble of rate $R_1 > 0$ and any number R_2 satisfying $R_2 < R_1 < 1$, there exists an ensemble of punctured LDPC code of rate R_1 and parent rate R_2 with the same performance. We also propose a method to construct the punctured code with the same performance as a given code. Although these punctured codes have the same asymptotic performance as the unpunctured ones, they can have better finite-length performance.

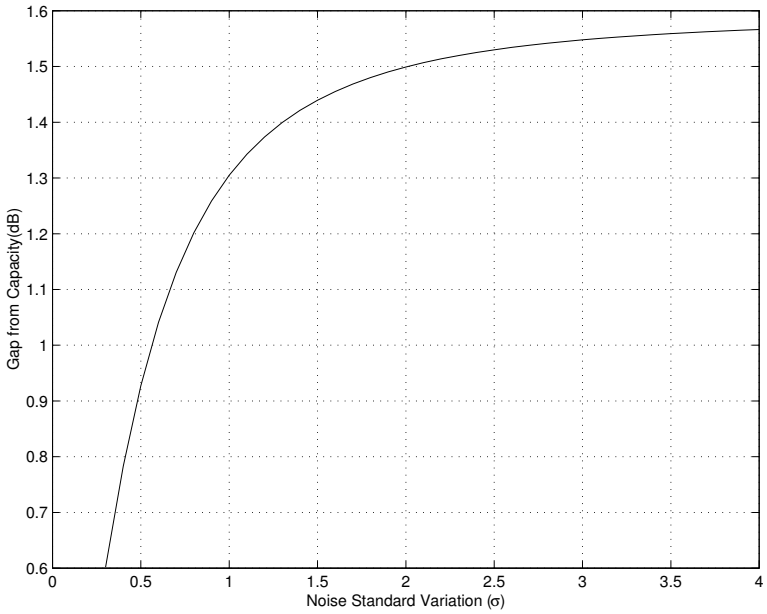

Figure 39: The gap from the capacity for an ensemble of punctured LDPC codes over BIAWGN channel.

Consider the parity check equation

$$c_1 : x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 = 0 \tag{180}$$

By splitting the above parity-check equation, we get

$$\begin{aligned} c_2 : x_1 \oplus x_2 \oplus x_3 \oplus y = 0 \\ c_3 : x_4 \oplus x_5 \oplus y = 0 \end{aligned} \tag{181}$$

where we refer to y as an augmented variable node. Figure 40 shows the effect of the parity-check splitting on the Tanner graph of the code. Note that this is different from the splitting operation introduced in [63]. Now consider an LDPC code in which some of the parity-check nodes have been split. This code can be considered as a punctured code. For example, the variable node y is a punctured variable node in Figure 40. By splitting process we can make a graph corresponding to a lower-rate code. When we puncture this code, we get a code with the same rate as the original code. Note that the splitting can be performed repeatedly and a check node that is obtained by splitting can itself be split to more check nodes. Therefore, we can have arbitrarily small rates. However, we assume that any check node is split only a finite number of times.

Consider an ensemble (λ, ρ) of LDPC codes of rate R_1. Let (λ, ρ, R_2) be the ensemble of codes obtained by splitting some of the check nodes of a code from the ensemble (λ, ρ) such that the rate of the code is decreased to R_2. Note that R_2 is the rate of the unpunctured codes. Obviously, puncturing the augmented variable nodes from a code in the ensemble (λ, ρ, R_2) results a code of rate R_1. For a graph g in the ensemble (λ, ρ), we define the graph g_{sp} to be the corresponding graph in (λ, ρ, R_2). For a check node c in a graph from the ensemble (λ, ρ), we define $Sp(c)$ as follows. If c is not split in g_{sp} then $Sp(c) = c$. Otherwise $Sp(c)$ is defined to be the set of check nodes in g_{sp} obtained by splitting c. For instance, in the above example we have $Sp(c_1) = \{c_2, c_3\}$. Similarly, for a set of check nodes A, we define

$$Sp(A) = \bigcup_{c \in A} Sp(c) \tag{182}$$

Lemma 16. *If we puncture all the augmented bits from the ensemble (λ, ρ, R_2), then the resulting ensemble has the same threshold as the ensemble (λ, ρ) under the message passing decoding algorithm.*

Proof. Let v be a variable node in a graph g from the ensemble (λ, ρ). Let also $P_v^{(l)}$ be the probability that the estimate of the variable node v in the l'th iteration of the message passing algorithm be wrong. Define $P_v^{'(l)}$ to be the corresponding probability when the decoding is performed on g_{sp}. Let $C_v^{(l)}$ and $V_v^{(l)}$ be respectively the sets of check nodes and variable nodes in the neighborhood of v that affect $P_v^{(l)}$. Define the sets $C_v^{\prime(l)}$ and $V_v^{\prime(l)}$ accordingly.

As it is shown in [112], with high probability the neighborhood of v (of constant depth) is tree-like. Therefore, $P_v^{(l)}$ is equal to the error probability of the maximum-likelihood estimation of v given the check nodes in $C_v^{(l)}$ and initial LLR's of the variable nodes in $V_v^{(l)}$. It is easy to show that the neighborhood of v in g_{sp} is tree-like as well. Choose $l' < \infty$ large enough such that $Sp(C_v^{(l)}) \subseteq C_v^{'(l')}$ and $V_v^{(l)} \subseteq V_v^{'(l')}$. Therefore, $P_v^{'(l')}$ is equivalent to the error probability of the ML-estimator of v given more information than what is provided by $C_v^{(l)}$ and $V_v^{(l)}$. Hence, $P_v^{'(l')} \leq P_v^{(l)}$. Thus, if the average error probability under the message passing decoding on the graph g tends to zero as l goes to infinity, the same thing should

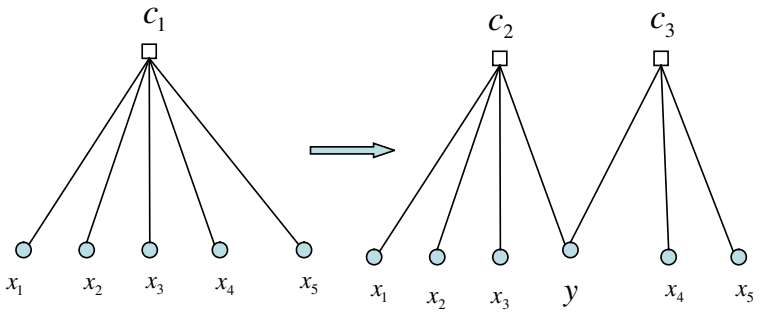

Figure 40: Splitting a parity check equation.

be true for g_{sp}. Conversely, it can be shown that if the average error probability under the message passing decoding on the graph g_{sp} tends to zero as l goes to infinity, the same thing should happen for g. Thus, we conclude that the thresholds of the two ensembles are the same. □

An immediate result of Lemma 16 is the following theorem [103, 103].

Theorem 24. *For any ensemble (λ, ρ) of LDPC codes of rate $R_1 > 0$, and any number R_2 satisfying $R_2 < R_1 < 1$, there exists an ensemble of punctured LDPC code of rate R_1 with parent rate R_2 having the same threshold under the belief propagation algorithm.*

Theorem 24 implies that if we design the codes properly, punctured codes are as good as ordinary LDPC codes. It also shows how to construct a punctured code with the same performance as the unpunctured code.

It is worth noting that although Lemma 16 states that the graphs g and g_{sp} have the same asymptotic thresholds, the two codes can have different finite-length performance. In fact, by a suitable choice of the punctured variable nodes, we may be able to alleviate the destructive effect of short cycles in the Tanner graph. When the code length is short, the short cycles of the graph deteriorate the performance. Using splitting, we can increase the cycle lengths and this can improve the performance of the finite-length codes.

6.2.4 Puncturing over the Binary Erasure Channel

For the erasure channel stronger results can be obtained. For example, using only one encoder and decoder we can achieve the capacity of BEC on arbitrary set of rates. As

another example, a stronger result than Lemma 16 is valid. Note that Lemma 16 holds only for the asymptotic threshold of the codes. We now show that for any graph g, the error probability of the codes corresponding to g and g_{sp} are the same even for finite-length codes. Let us define the merging of two check nodes that are connected to a punctured node of degree two as the reverse operation of splitting. For instance, in the above example if we replace the two check nodes c_2 and c_3 by c_1 and delete the vertex y from the graph, we say we have merged c_2 and c_3.

Lemma 17. *Let g be a bipartite graph with bipartition $V(g)$ and $C(g)$, the set of variable nodes and check nodes, respectively. Let also g_{sp} be a bipartite graph obtained by splitting some check nodes in g. Define $V(g_{sp})$ as the set of variable nodes in g_{sp}. We write $V(g_{sp}) = V_{sp} \cup V_p$, where $V_{sp} = V(g)$ and V_P is the set of augmented variable nodes. A set $S \subseteq V(g)$ is a stopping set in g if and only if there exists a set $U \subseteq V_p$ such that $S \cup U$ is a stopping set in g_{sp}.*

Proof. Let $S \subseteq V(g)$ be a stopping set in g. Let $N_{g_{sp}}(S)$ be the set of neighbors of S in g_{sp} (i.e., the set of parity-check nodes in g_{sp} that are connected to some variable nodes in S). If for any $c \in N_{g_{sp}}(S)$ we have $deg_S(c) \geq 2$ then S is a stopping set in g_{sp}, as well. Otherwise, there exists a parity-check equation $c_1 \in N_{g_{sp}}(S)$ with $deg_S(c_1) = 1$. Since S is a stopping set in g, c_1 must have been split from a check node c in g. Suppose c has been split to $c_1, c_2, ..., c_j$. Since S is a stopping set in g, at least one of the check nodes $c_2, ..., c_j$ has a neighbor in S. Suppose c_t is connected to the variable node w in S. Thus, there is a path $c_1 - v_1 - c_2 - v_2 - ... - v_{t-1} - c_t - w$ in which $v_1, v_2, ..., v_{t-1}$ are augmented nodes of degree two. Thus $S \cup \{v_1, v_2, ..., v_{t-1}\}$ is a stopping set in g_{sp}.

Now suppose the sets $S \subseteq V(g) = V_{sp}$ and $U \subseteq V_p$ be such that the set $S \cup U$ is a stopping set in g_{sp}. Let C_s be the set of check nodes in $N_{g_{sp}}(S)$ that have a neighbor in U. We merge these check nodes to the original check nodes in g. For instance, suppose we merge the check nodes $c_1, c_2, ..., c_j$ to get a new check node c. We now show that c has at least two neighbors in S. Let G' be the graph induced by the nodes in $S \cup U$ and their neighbors in g_{sp}. Since c had been split to the check nodes $c_1, c_2, ..., c_j$, there is a path

$c_1 - v_1 - c_2 - v_2 - ... - v_{j-1} - c_j$ in G' where $v_1, v_2, ..., v_{j-1}$ are the augmented nodes of degree two. However, the degrees of c_i's are at least two in the graph G'. Therefore, both c_1 and c_j must have neighbors in S. Thus, the check node c must have at least two neighbors in S. Thus we conclude that S is a stopping set in g. □

The immediate result of Lemma 17 is the following theorem [100].

Theorem 25. *Let g be a bipartite graph and g_{sp} be a bipartite graph obtained by splitting some check nodes in g. Now suppose we puncture all the augmented variable nodes from the graph g_{sp}. The resulting code has the same bit error probability (computed for only unpunctured bits) as the code that corresponds to g over the erasure channel under standard iterative decoding.*

In the previous section, we stated a general result regarding puncturing in Theorem 22. Here, we show stronger results for the BEC. For example, random puncturing of a code over BEC results in no performance loss. Using this fact we can show that for any R, $0 < R < 1$, it is possible to design a code of rate R with the following property. The code is capacity achieving if it is randomly punctured to any rate $R_1 \geq R$.

Lemma 18. *Let C_1 be an LDPC code of rate R_1 and length n. Let e_1 be the bit error rate of the standard iterative decoding of the code when used over a BEC with erasure probability ϵ_1. Consider the ensemble (C_1, p) of LDPC codes that is obtained by randomly puncturing the code C_1 with puncturing fraction p. Choose ϵ_2 such that*

$$\frac{1-\epsilon_1}{R_1} = \frac{1-\epsilon_2}{R_2} \tag{183}$$

where $R_2 = \frac{R_1}{1-p}$ is the rate of the ensemble (C_1, p). Let e_2 be the average bit error rate of a randomly chosen code from the ensemble (C_1, p) over a BEC with erasure probability ϵ_2. Then we have

$$e_1 = e_2. \tag{184}$$

Proof. This lemma is a special case of Theorem 3 in [106]. □

It is also easy to show that the bit error rate of a randomly chosen code from the ensemble (C_1, p) is highly concentrated about the average value using the same arguments as in [70] and [112]. Now we can state the following theorem [100, 103].

Theorem 26. *Let $T \subseteq (0,1)$ with $\inf_{R \in T} R > 0$ and δ be any positive constant. Then, there exists an ensemble (λ, ρ) of LDPC codes with the following property. The ensemble (λ, ρ) can be punctured randomly to get an ensemble of the arbitrary rate $R \in T$ such that*

$$\forall\ R \in T; R \geq (1-\delta)c_R. \tag{185}$$

Here, $c_R = 1 - \epsilon_{th}(R)$ and $\epsilon_{th}(R)$ is the threshold of the punctured ensemble of rate R under the standard iterative decoding.

Proof. Let $R_p = \inf_{R \in T} R$ and let $\{(\lambda_N, \rho_N)\}$ be a sequence of capacity achieving degree distributions of rate R_p [70], [118], [119] and [84]. Choose N large enough such that $R_p \geq (1-\delta)c_{R_P}$. Since

$$\frac{c_{R_p}}{R_p} = \frac{c_R}{R}$$

for all $R \in T$, by Lemma 18 we conclude the proof. □

In other words, when we have a capacity achieving sequence of LDPC codes of rate R, the ensemble remains capacity achieving when it is punctured to a higher rate. Thus, we can design only one encoder and decoder and obtain arbitrary many rates. Moreover, the code is capacity achieving for all the desired rates over the BEC.

6.2.5 Design of Good Punctured LDPC Codes

In this section we discuss the design of good rate-compatible LDPC codes using puncturing. As it was shown in the previous section, design of punctured LDPC codes over the BEC is very simple. We just need to use a good degree distribution for the parent code and the punctured performs very well for all higher rates. On the other hand, it is not obvious how to design good puncturing schemes for other channels .

We first note that if the desired range of rates is short, then we can choose a good code for the smallest rate and randomly puncture it to get codes of higher rates. Simulations shows

that this simple structure is practically efficient. Our experience shows that increasing the code rate by an amount less than forty percent is usually obtained by a very small performance loss. As an example, the diagrams in [46], suggest that when the rate increase is less than forty percent, the performance loss is less than .2 dB for the BIAWGN channel. This is true even when we are using random puncturing. Thus we may focus on the cases that a broad range of rates is needed, specifically when rates close to 1 are needed.

A necessary condition is to choose a code with high enough puncturing threshold. That is, if R_m and R_p are the highest desired rate and the parent code rate, respectively, and p_{th} is the puncturing threshold of the parent code, we must have $p_{th} > 1 - \frac{R_p}{R_m}$. Here we propose a technique that works well in practice.

As we mentioned previously a punctured code can be viewed as Figure 37. Hence, we can consider the puncturing process as variation in the channel not in the code rate. From the previous section we know that the highest rate that we need plays an important role in the performance of the punctured code. When the puncturing fraction is maximum, the channel is close to a BEC. For the random puncturing, the resulting binary erasure channel is assumed to be uniform but for the intentional puncturing, the BEC is assumed to be non-uniform. A simple method is to choose the parent code to be a good code for BEC. By this choice, we expect to get good performance at the highest rates. However, as it is discussed in [24], with a little care, the code that is optimized over BEC is also optimal over other MBIOS channels. Thus we expect to get good performance even at very low rates. In fact, our experience shows that the most destructive problem of the punctured codes is the threshold effect. If the gap between the highest rate and the cut off rate is not enough, large performance degradation occurs at high rates.

In order to examine the above methodology we chose a good ensemble of half-rate LDPC codes in [1] with the following degree distribution

$$\lambda_5(x) = 0.2498x + .2472x^2 + .1480x^5 + .0033x^6 + .3517x^{19}, \tag{186}$$
$$\rho_5(x) = x^7.$$

The ensemble has the cut off rate of $R_c = .9797$. We generate an LDPC code of length 10^5 from this ensemble. To compare our results with [46], we measured the gap to Shannon limit of this code at the bit error rate of 10^{-4}. Figure 41 shows the simulation results for this code when it is randomly punctured to generate the rates at the range of .5 to .91. We note that for all rates the gap from the capacity is less than $.7dB$. To compare the performance of this code to the codes given in [46], we examine the performance of two half-rates codes in [46] that are punctured to higher rates. We note that both codes show about $1.8dB$ gap to the capacity at rate .91. Even with optimized intentional puncturing the codes have $1dB$ gap to capacity at this rate. We also note that our code has a smaller length than those in [46].

It is worth noting that random puncturing is more suitable than intentional puncturing for rate-compatible coding. This is because we choose a fraction p_1 of the variable nodes at random for the first rate. For the next rate, we choose more bits at random from the unpuncured bits and so on. Thus, we do not need optimization for puncturing, reduce the degradation at higher rates, and do the puncturing in a rate-compatible way.

An important property of the above scheme is that it is extendable to finite-length codes. Using recent breakthrough in the design and analysis of finite-length LDPC codes over the BEC [27], [109], and [111] we can find good LDPC codes over the BEC and design efficient punctured LDPC codes.

6.3 Capacity Achieving Sequences for MBIOS Channels Using Punctured codes

It has been conjectured that for any MBIOS channel there exists a sequence of capacity achieving LDPC codes with iterative decoding, see for example [54]. Although this has

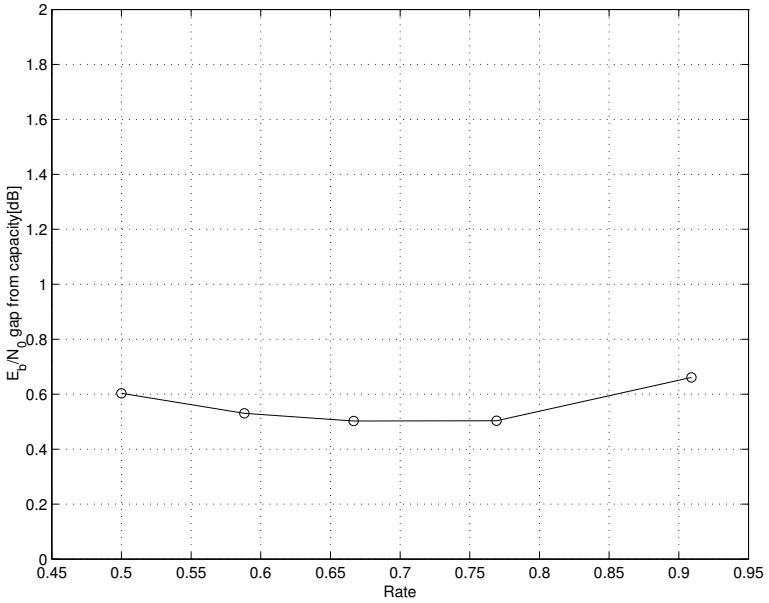

Figure 41: The gap from capacity for a randomly punctured LDPC code of length 10^5 chosen from the ensemble (λ_5, ρ_5) at the bit error rate of 10^{-4}.

been one of the most important open problems in coding theory, it has been proven only for the BEC. In this section we show that punctured LDPC codes may be helpful to verify the conjecture. We show that if the conjecture is proved for the rates approaching zero, then it will be valid for all rates.

Consider the class of MBIOS channels parameterized by a parameter θ with capacity c_θ. If the threshold of an ensemble of codes under the message passing decoding is θ_{th}, we say that the capacity of the ensemble is $c_{\theta_{th}}$. We say that a sequence of degree distributions $\{(\lambda_n, \rho_n)\}_{n=1}^{\infty}$ with rate R is capacity achieving if for any positive constant δ, there exists an integer N such that if $n > N$, then $R \geq (1-\delta)c_{\theta_{th_n}}$. Here $c_{\theta_{th_n}}$ is the capacity of the ensemble (λ_n, ρ_n). We now suggest the following research problem.

Research Problem 1. *For any class of MBIOS channels, there exists a sequence of degree distributions* $\{(\lambda_n(R), \rho_n(R))\}_{n=1}^{\infty}$ *such that*

$$\lim_{R \to 0} \frac{R}{c(R)} = 1 \tag{187}$$

where $c(R)$ *is the capacity of the ensemble* $(\lambda_n(R), \rho_n(R))$*, when* n *tends to infinity.*

We now show that if the above research problem is proved, then capacity achieving

LDPC codes of all rates exist. The important point is that it might be easier to prove the existence of the capacity achieving sequences for the case where the rate of the code is approaching zero. This is because, for this case, the channel is approaching a channel with zero capacity and all different MBIOS channels may become somewhat similar. For example, one approach to solve the Research Problem 1 is using the capacity achieving sequences for the BEC as their rates approaches zero.

Theorem 27. *[100, 103] If the statement of Research Problem 1 is true, then there exists a capacity achieving sequence of LDPC codes for any MBIOS channel.*

Proof. Let C_θ be a MBIOS channel with capacity c_θ and let δ be any positive constant. Consider the channel C_{eq} in Figure 37. The capacity of C_{eq} is equal to $c_{eq}(p) = (1-p)c_\theta$, where p is the puncturing fraction. Suppose we are using random puncturing. The channel $C_{eq}(p)$ can be considered as a MBIOS channel with the parameter p. When p approaches one the capacity of $C_{eq}(p)$ tends to zero. Thus, by Research Problem 1, there exists a sequence of degree distributions $\{(\lambda_n(R), \rho_n(R))\}_{n=1}^{\infty}$ such that

$$\lim_{p\to 1} \frac{R}{c_{eq}(p)} = 1. \tag{188}$$

This code can be used for reliable communication over $C_{eq}(p)$. Thus for large enough n and p, we have an ensemble of LDPC codes of rate $R \geq (1-\delta)c_{eq}(p)$. This ensemble can be considered as a punctured ensemble with effective rate $R_{eff} = R/(1-p)$. Therefore, we have

$$R_{eff} = \frac{R}{1-p} \geq \frac{(1-\delta)c_{eq}(p)}{1-p} = (1-\delta)c_\theta \tag{189}$$

This implies that the punctured code with rate $R_{eff} \geq (1-\delta)c_\theta$ can be used for reliable communication over C_θ. □

Note that Theorem 27 is quite general and can be applied to any code ensemble. In fact, the proof does not require to consider LDPC codes.

6.4 Raptor Codes

Raptor codes were introduced by Shokrollahi [117] for the erasure channel. The applications of these codes on general symmetric channels was studied in [34] and [85]. Raptor codes

are used as rate-less codes. In these applications we may not know the channel statistics or the statistics may be time variant. For a an information block of length k, raptor codes produce a potentially infinite stream of symbols. It is shown in [117] that if the raptor code is designed properly for any $\epsilon > 0$, any subset of size $k(1+\epsilon)$ of the generated symbols is sufficient to recover the original k symbols with high probability. In other words, raptor codes are capacity achieving over the erasure channel independent of the channel erasure probability. This property is called universality of raptor codes on the erasure channel. The application of these codes over other symmetric channels has been studied in [34] and [85]. It is shown in [34], that raptor codes are not universal over symmetric channels by showing that the required fraction of degree-two output bits depends on the noise level of the channel. In this section we introduce a construction of raptor codes that circumvent this problem in certain scenarios by allowing different distribution for different output bits.

Here we consider the following scenario. We assume that all symbols that are generated by the encoder are transmitted through a memory-less binary-input output-symmetric (MBIOS) channel and are received at the receiver in the same order that have been sent to the channel. We will show that in the above scenario we can use generalized Raptor codes to approach the channel capacity in a rate-compatible way. It is worth noting that in the original scenario of Raptor codes it is assumed that the receiver obtains only a subset of symbols sent by the sender. This is specially a reasonable assumption in the networking applications that the channel is modeled as a BEC. However, in many applications in which rate-compatible coding is required our assumption that the receiver gets all the corrupted symbols is practical.

Consider an MBIOS channel with parameter θ, where $\theta \in [\theta_{min}, \theta_{max}]$ and $\theta_{min}, \theta_{max} \in \mathbb{R} \cup \{-\infty, +\infty\}$. For example, for the binary-input additive white Gaussian noise (BIAWGN) channel, θ can be considered as the variance σ of the noise. Let $\mathcal{C}$ be a class of channels with parameter θ. Thus, any channel $\mathcal{C}_\theta$ in $\mathcal{C}$ in is uniquely determined by its variable θ. A channel in $\mathcal{C}$ with parameter θ_0 is called $\mathcal{C}_{\theta_0}$. The capacity of the channel $\mathcal{C}_{\theta_0}$ is shown by Cap $(\mathcal{C}_{\theta_0})$. Similar to [112], we consider physically degraded channels. For clarity of exposition we assume that if $\theta_1 < \theta_2$, then $\mathcal{C}_{\theta_2}$ is physically degraded with respect to

$\mathcal{C}_{\theta_1}$. Let X and Y be the random variables representing the input and output of a MBIOS channel, respectively. For the channel $\mathcal{C}_{\theta_0}$ we define the random variable Z_{θ_0} in the following way. We let the input to the channel be $X = 1$. If y is the output of the channel, then

$$Z_{\theta_0} = \log \frac{\text{pr}\{X = 1|Y = y, \theta = \theta_0\}}{\text{pr}\{X = -1|Y = y, \theta = \theta_0\}}, \tag{190}$$

Let $F_0(x;\theta)$ be the distribution function of Z_θ. The capacity of the normalized channel is given by [24]

$$\text{Cap}(\mathcal{C}_{\theta_0}) = 1 - E[\log_2(1 + e^{-Z_{\theta_0}})|X = 1]. \tag{191}$$

Another useful parameter which is defined in [34] is $E(\mathcal{C}_{\theta_0})$,

$$\text{E}(\mathcal{C}_{\theta_0}) = E[\tanh(\frac{Z_{\theta_0}}{2})|X = 1]. \tag{192}$$

6.4.1 Conventional Raptor Codes

We now briefly describe the construction of raptor codes introduced in [117] and [34]. Let k be a positive integer. Let $\Omega_1, \Omega_2, ..., \Omega_k$ be a distribution on $\{1, 2, ..., k\}$ so that Ω_i denotes the probability that the value i is chosen. The distribution generator polynomial is given by $\Omega(x) = \sum_{i=1}^{k} \Omega_i x^i$. Let C be a linear code of block-length n and dimension k. A raptor code with parameters $(k, C, \Omega(x))$ is an LT-code [69] with distribution $\Omega(x)$ on n symbols which are the coordinates of codeword C. In other words, the encoding is done in the following way. The k information symbols are encoded using the code C, so that we obtain n intermediate symbols. Then the output symbols are generated from these n bits using an LT-code that has distribution $\Omega(x)$. That is for each symbol an integer w is chosen based on the distribution $\Omega_1, \Omega_2, ..., \Omega_k$. Then a subset of n intermediate symbols is chosen randomly from all possible subsets of weight w. The output symbol is obtained by adding the symbols in this subset. For simplicity here we consider binary symbols, so the addition is equivalent to the exclusive OR operation. Code C is usually chosen to be an LDPC code, so that the decoding can be done using message passing algorithms on the resulting Tanner graph.

The decoder works as follows. The receiver receives the output symbols that have been transmitted through the channel. Once the receiver gets a sufficient number of symbols

from the channel, it tries to decode the original k bits. If the decoding fails, the receiver gets more symbols from the channel and attempts decoding again.

It has been shown in [117] that Raptor codes achieve the capacity of the BEC for all rates. The existence of capacity achieving Raptor codes for other symmetric channels has not been proved yet. However, even if we assume capacity achieving Raptor codes exist for other channels, they cannot achieve the capacity for all rates, i.e, they are not universal. In particular, the following result has been proved in [34].

Theorem 28. *Suppose that $(k_m, C_m, \Omega^{(m)}(x))$, $m \geq 1$, is a capacity-achieving sequence of Raptor codes for the channel $\mathcal{C}_\theta$ and suppose that Cap $(\mathcal{C}_{\theta_0}) \neq 0$. Then we have*

$$\lim_{m\to\infty} \Omega_1^{(m)} = 0, \quad \lim_{m\to\infty} \Omega_2^{(m)} = \frac{Cap(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)}. \tag{193}$$

For the binary erasure channel (BEC), we have $\frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)} = \frac{1}{2}$. However, for general symmetric channels the quantity $\frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)}$ depends on the channel parameter. Thus, it is not possible to construct universal Raptor codes for symmetric channels. In the next section we will show that this problem can be solved by allowing a variable distribution. We call the resulting codes, generalized Raptor codes.

6.4.2 Generalized Raptor Codes

Here we introduce generalized Raptor codes that can be used over general symmetric channels. The main point can be described as follows. For the BEC, the capacity-achieving distributions do not depend on the code rate. Thus, by one distribution we can achieve the capacity for all rates. However, for other symmetric channels, the capacity achieving distributions (if there exist such distributions) depend on the channel capacity by Theorem 28. To fix this problem we change the distribution as we change the rate. For simplicity of presentation we assume that we want to work at only two rates. All discussions can be trivially extended to arbitrary set of rates. In particular suppose we want to have two rates R and R', where $0 < R' < R < 1$. Also suppose θ and θ' are the corresponding channel parameters. That is, $\text{Cap}(\mathcal{C}_\theta) = R$ and Cap $(\mathcal{C}_{\theta'}) = R'$. Similar to ordinary raptor codes the original k bits are first encoded to n intermediate bits using the pre-code C. The

first $\frac{n}{R}$ output bits are encoded using distribution $\Omega(x)$. The distribution $\Omega(x)$ is chosen to guarantee that the raptor code good performance for rate R. The next $\frac{n}{R'} - \frac{n}{R}$ output bits are encoded using another distribution $\Omega'(x)$. The distribution $\Omega'(x)$ is chosen such that the overall distribution $\Omega''(x) = \frac{R'}{R}\Omega(x) + (1 - \frac{R'}{R})\Omega(x)$ is a good distribution for rate R'. In this way, when the decoder receives the first $\frac{n}{R}$ corrupted symbols from the channel, it can attempt the decoding. If the channel capacity is large enough, the decoding is successful, otherwise the decoder gets the remaining $\frac{n}{R'} - \frac{n}{R}$ symbols and attempts the decoding again. It seems reasonable that the generalized Raptor codes should outperform the ordinary Raptor codes; however, it is not clear that these code can achieve the capacity for all the desired rates. In the following we will show that the necessary condition of Theorem 28 can always be satisfied by generalized Raptor codes. Since the values of $\Omega_i, i > 2$ for a capacity-achieving distribution for symmetric channels is not known, we cannot verify that it is actually possible to achieve the capacity for all rates. However, it is clear that ordinary Raptor codes are a special case of generalized Raptor codes, so by using generalized Raptor codes we can only improve the performance. We first state the following result.

Lemma 19. *Consider the generalized Raptor code ensemble described above. The asymptotic threshold of the codes from the ensemble at rate R' is the same as ordinary Raptor codes $(k, C, \Omega''(x))$, where $\Omega''(x) = \frac{R'}{R}\Omega(x) + (1 - \frac{R'}{R})\Omega(x)$.*

Proof. The proof is straightforward and is done by considering the density evolution formulas. It is easily seen that both ensembles have the same threshold. □

Consider a symmetric channel $\mathcal{C}_\theta$ with capacity Cap $(\mathcal{C}_\theta)$ as described above. Moreover, assume that $E(\mathcal{C}_\theta)$ and $\frac{1}{2E(\mathcal{C}_\theta)} - \frac{1}{\text{Cap}(\mathcal{C}_\theta)}$ are non-decreasing with respect to channel capacity. It is easy to verify almost all practical symmetric channels such as binary symmetric channel (BSC) and BIAWGN channel satisfy this property. We now prove the following result. For the BSC and BIAWGN the result has been stated in [34], here we prove it for general $\mathcal{C}_\theta$. For a symmetric channel $\mathcal{C}_\theta$, let $\Omega_2(\mathcal{C}_\theta) = \lim_{m\to\infty} \Omega_2^{(m)} = \frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)}$ be the required fraction of degree-two output nodes to achieve the capacity.

Lemma 20. *For a symmetric channel $\mathcal{C}_\theta$, we have*

$$\Omega_2(\mathcal{C}_\theta) = \frac{Cap(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)} \leq \frac{1}{2}. \tag{194}$$

Proof. It suffices to show $\text{Cap}(\mathcal{C}_\theta) \leq E(\mathcal{C}_\theta)$. In other words we have to prove $1 \leq E[\log_2(1+e^{-Z_\theta})|X=1] + E[\tanh(\frac{Z_\theta}{2})|X=1]$. We have

$$\begin{aligned}
&E[\log_2(1+e^{-Z_\theta})|X=1] + E[\tanh(\frac{Z_\theta}{2})|X=1] \\
&= \int_{-\infty}^{+\infty} \Big(\log_2(1+e^{-z}) + \tanh(\frac{z}{2}) \Big) dF_0(z;\theta) \\
&= \int_{0}^{+\infty} \Big(\log_2(1+e^{-z}) + \tanh(\frac{z}{2}) + e^{-z}[\log_2(1+e^{z}) + \tanh(\frac{-z}{2})] \Big) dF_0(z;\theta) \\
&= \int_{0}^{+\infty} \frac{\Big(\log_2(1+e^{-z}) + \tanh(\frac{z}{2}) + e^{-z}[\log_2(1+e^{z}) + \tanh(\frac{-z}{2})] \Big)}{(1+e^{-z})} (1+e^{-z}) dF_0(z;\theta) \\
&\geq \int_{0}^{+\infty} 1.(1+e^{-z}) dF_0(z;\theta) = 1,
\end{aligned}$$

where we have used the channel symmetry and the fact that $\log_2(1+e^{-z}) + \tanh(\frac{z}{2}) + e^{-z}[\log_2(1+e^{z}) + \tanh(\frac{-z}{2})] \geq (1+e^{-z})$, for $z \geq 0$. □

We now prove that for a symmetric channel $\mathcal{C}_\theta$, we can always satisfy the necessary condition of Theorem 28 using generalized Raptor codes.

Theorem 29. *Suppose that $(k_m, C_m, \Omega^{(m)}(x))$, $m \geq 1$, is a sequence of Raptor codes for the channel $\mathcal{C}_\theta$ that satisfies*

$$\lim_{m\to\infty} \Omega_2^{(m)} = \frac{Cap(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)}. \tag{195}$$

Then by suitably choosing $\Omega_2^{'(m)}$ in $\Omega'(x)$, the equivalent Raptor code at rate R' satisfies

$$\lim_{m\to\infty} \Omega_2^{''(m)} = \frac{Cap(\mathcal{C}_{\theta'})}{2E(\mathcal{C}_{\theta'})}. \tag{196}$$

Proof. We have $\Omega''(x) = \frac{R'}{R}\Omega(x) + (1-\frac{R'}{R})\Omega(x)$, thus $\Omega_2^{''(m)} = \frac{R'}{R}\Omega_2^{(m)} + (1-\frac{R'}{R})\Omega_2^{'(m)}$. Thus all values in the interval $[\frac{R'}{R}\Omega_2^{(m)}, \frac{R'}{R}\Omega_2^{(m)} + (1-\frac{R'}{R})]$ are achievable for $\Omega_2^{''(m)}$. Since

$\lim_{m\to\infty} \Omega_2^{(m)} = \frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)}$, it suffices to prove that

$$\frac{R'}{R}\frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)} \leq \frac{\text{Cap}(\mathcal{C}_{\theta'})}{2E(\mathcal{C}_{\theta'})} \leq \frac{R'}{R}\frac{\text{Cap}(\mathcal{C}_\theta)}{2E(\mathcal{C}_\theta)} + (1 - \frac{R'}{R}). \tag{197}$$

Considering $\text{Cap}(\mathcal{C}_\theta) = R$ and $\text{Cap}(\mathcal{C}_{\theta'}) = R'$, this can be concluded from the assumption that $E(\mathcal{C}_\theta)$ and $\frac{1}{2E(\mathcal{C}_\theta)} - \frac{1}{\text{Cap}(\mathcal{C}_\theta)}$ are non-decreasing with respect to channel capacity. □

Theorem 29 shows that generalized Raptor codes can satisfy the necessary condition of Theorem 28.

6.4.3 Simulation Results

We now show by an example that generalized Raptor codes can improve the performance. We consider the BIAWGN channel. We choose $k = 1000$, and use the following degree distribution that is given by [117]. We choose $D = 7$.

$$\Omega_D(x) = \frac{1}{\mu+1}\left(\mu x + \frac{x^2}{1.2} + \frac{x^3}{2.3} + ... + \frac{x^D}{(D-1).D} + \frac{x^{(D+1)}}{D}\right). \tag{198}$$

For the outer code, we use an LDPC code of length $n = 1200$ with the following degree distribution

$$\lambda(x) = \frac{2}{17}x + \frac{5}{17}x^2, \quad \rho(x) = x^{16}. \tag{199}$$

We consider two cases. First we use ordinary Raptor codes and evaluate the performance of the code as the noise level of the channel changes. Next, we use generalized Raptor codes. The first n output symbols are generated using the distribution $\Omega_D(x)$. However, for higher rates we try to optimize the distribution. Since, the code length is short we cannot use density evolution to optimize the distribution. Thus, we tried to find the best performance by generating the graphs based on a given distribution and finding the corresponding bit error rate. Figure 42, shows the performance of both methods. We see that we can obtain some improvements by using generalized Raptor codes.

6.5 Conclusion

We studied two methods for constructing rate-compatible codes. We first considered punctured LDPC codes. We studied some fundamental properties of punctured LDPC codes.

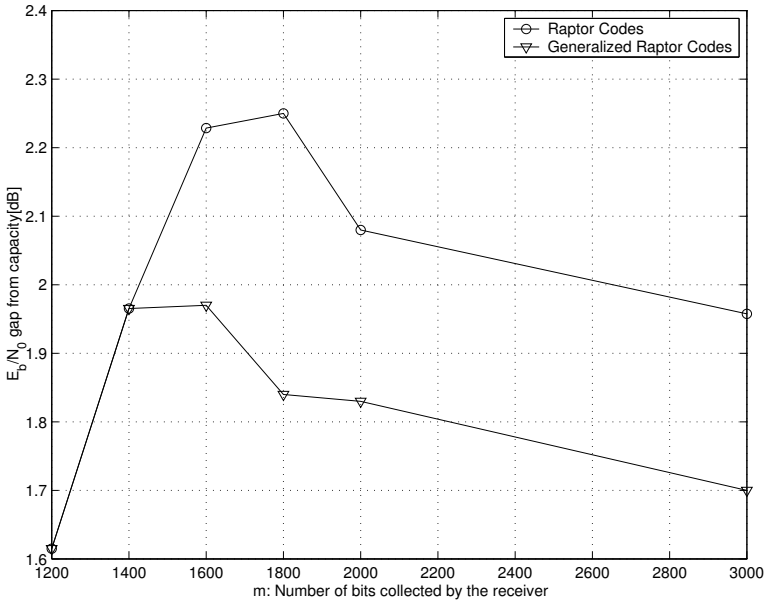

Figure 42: Gap from capacity for ordinary and generalized Raptor codes of length $k = 10^3$ at the bit error rates of 10^{-4} over the BIAWGN channel.

The threshold effect and the optimality of punctured LDPC codes were discussed. We showed that for any ensemble of LDPC codes, there exists a cut off rate which is the maximum achievable rate using puncturing. We proved the existence of asymptotically good punctured LDPC codes for an arbitrary range of rates. For the binary erasure channel, we find that using only one encoder and decoder, we can achieve the capacity of the channel over an arbitrary set of rates. We also proposed a simple method for designing rate-compatible LDPC codes that has several advantages over the previous methods. First, it reduces the performance degradation at high rates. Second, it is applicable to finite-length codes. Third, there is no need for optimizing the puncturing pattern. Forth, the puncturing can be done in a rate-compatible way. Finally, we showed that puncturing might help to solve an important open research problem. That is, the capacity achieving property of LDPC codes over MBIOS channels under the message passing decoding algorithms.

Finally, we studied Raptor codes. We introduced a construction of raptor codes for symmetric channels. An important property of the proposed scheme is that unlike the previous construction of Raptor codes, it allows to design codes that are approaching the capacity of the underlying channel in a rate-compatible way.

CHAPTER VII

CONNECTIVITY PROPERTIES OF LARGE-SCALE WIRELESS SENSOR NETWORKS

7.1 Introduction

Wireless sensor networks are emerging as a new technology with advances in both MEMS technology and networking. They have received a great deal of interest lately, with potential applications in military and civilian surveillance and sensing tasks and potential services that would enhance the ability of the growing domain of wireless technologies [4].

Wireless sensor networks consists of a large number (in the order of thousands) of identical nodes which are constrained in available energy, computational power, memory, and communication range. In potential sensing applications, the sensor nodes may be randomly deployed in a hazardous or dangerous environment where the nodes are physically inaccessible after deployment. Hence, the design of the network needs to consider energy conserving schemes to account for a limited energy supply, low memory/computation and resilient networking schemes to account for the hostile environment.

Resiliency in large-scale sensor networks is often linked to the connectivity of the network. That is, every node in the network should be able to communicate with the base stations in the network. Without such connectivity, the network is unable to provide proper functionality. Moreover, redundancy that is added through sending information through multiple paths is another characteristic within sensor networks that is utilized.

Graph theoretic properties of wireless networks have been studied extensively. In this chapter we consider the effect of node and link failures, which are common in sensor networks, on different network properties. Our model of sensor networks assumes n sensors are distributed randomly over a field based on a given distribution function. We include link failures in our model, that is two active sensor nodes are connected with probability $p_e(n)$

if they are within the communication range of each other. The parameter $p_e(n)$ represents the effect of link failures, that is a link fails with probability $1 - p_e(n)$. In sensor networks, different factors may contribute to link failures such as key management schemes. In random key management schemes [33], [20], two neighbor nodes can establish a link only if they share a key. In these schemes, we choose a random key pool from the key space. Each key has an identifier. Before deployment, each sensor node is given a random subset of keys along with their identifiers from the key pool. If two nodes are in the communication range of each other and share a common key identifier, then they can use the corresponding key as their shared secret to initiate communication. In [20], authors gave a modified version of the above scheme which they called q-composite key predistribution scheme. If $s(n)$ is the number of keys in the key pool and $k(n)$ is the number of keys stored in each sensor, then we have

$$p_e(n) = 1 - \frac{\binom{s(n)}{k(n)}\binom{s(n)-k(n)}{k(n)}}{\binom{s(n)}{k(n)}^2}. \tag{200}$$

Usually, $k(n)$ and $s(n)$ are chosen such that $p_e(n)$ is bounded away from zero as n grows [33], [20]. Node failure is also a common phenomenon in sensor networks. Sensor nodes may fail due to lack of power, physical damage or environmental interference [4]. It is very important that the network can still continue to work properly even after some nodes have failed. In our model any sensor node may fail with probability $1 - p_{sf}(x, y, n)$, where (x, y) is the location of the node in the plane. For simplicity we study the link failures and node failures separately. First, we study the effect of link failures on the network. While some properties of link-reliable networks (networks with reliable links) can be easily extended to networks with unreliable links, some other properties require more complicated analysis. We then study the effect of node failures. We prove general statements relating node-reliable networks to unreliable ones. Using this general theorems we study the properties of networks with unreliable sensor nodes. Finally, we show that the two results can be combined for the analysis of networks with unreliable nodes and links.

The focus of this chapter is to provide analysis of some network properties that affect network functionality. We study k-connectivity of large-scale sensor networks. We derive

necessary and sufficient conditions for k-connectivity of the network graph. We study the minimum communication radius of sensor nodes to provide k-connectivity within the network. We analyze the average shortest path of the k paths from a node in the sensing field back to a base station. We also study the existence of the giant component (a large subset of nodes that are connected). These results have been shown through graph theoretical derivations and also have been verified through simulations. For clarity of exposition, we provide the lengthy proofs in Appendix D. However, it should be noted that a major contribution of this chapter is to provide the mathematical methodology for dealing with large-scale sensor networks. Thus, an important part of the chapter lies in the proofs of the results given in Appendix D.

Formally, we say that a graph is connected if there is a path between every pair of vertices. A graph is said to be k-vertex-connected or simply k-connected if there does not exist a set of $k-1$ vertices whose removal disconnects the graph. For $k \geq 2$, we say a graph is k-edge-connected if it has at least two vertices and no set of at most $k-1$ edges separates it.

The k-connectivity property is important from the network reliability perspective. In particular, a k-connected network remains connected if less than k nodes are removed from the network as a result of node failures or an attack by an enemy. Moreover, k-connectivity is necessary for multi-path routing. The concept of k-connectivity considers a random graph and infers that there exists k disjoint paths between each pair of nodes. Thus, there exist k disjoint paths between any two nodes if an only if the associated random graph is k-connected. In terms of wireless networks, this implies that, on the link level, there exists k disjoint paths from each pair of nodes by hopping through unique sets of intermediate nodes. In the case of sensor networks, it is important to show the k-connectivity between the base station(s) and each sensor node in the field. However it is still up to the route discovery mechanism to find these k disjoint paths. The existence of the giant component is important when the network loses connectivity. In some applications, it is sufficient for the operation of the network to have a large subset of active nodes connected to each other (i.e., the network possesses a giant component).

Throughout the chapter we assume $\mathcal{B}(\mathbb{R}^2)$ is the Borel σ-algebra on $\mathbb{R}^2$ and m is the Lebesgue measure on $\mathcal{B}(\mathbb{R}^2)$. $\overline{B(\overline{X}, R)}$ is the closed ball with radius R centered at $\overline{X}$ in $\mathbb{R}^2$. $\overline{S(\overline{X}, L)}$ is the closed square with side L centered at $\overline{X}$ in $\mathbb{R}^2$. In particular $S_0 = \overline{S(\overline{O}, 1)}$ is the closed square with unit area centered at the origin. For any $E \in \mathcal{B}(\mathbb{R}^2)$ we define $\nu(E) = m(E \cap S_0)$. Clearly ν defines a measure on $\mathcal{B}(\mathbb{R}^2)$. For an integer n, $(n)_k = n(n-1)...(n-k+1)$. For a random variable Y, $E(Y)_k$ shows the k'th factorial moment. That is $E(Y)_k = E[Y(Y-1)...(Y-k+1)]$. Let ε_n be an event depending on a parameter n. We say that ε_n holds asymptotically almost surely, or ε_n holds with high probability, if $\Pr\{\varepsilon_n\}$ tends to 1 as $n \to \infty$. For two sequences a_n and b_n, $a_n \sim b_n$ means $\lim_{n\to\infty} \frac{a_n}{b_n} = 1$.

The remainder of the chapter is structured into several parts. The next section provides an overview of the work related to our study. Section 7.3 establishes the formulation and preliminaries of the problem we have considered. Section 7.4 studies sensor networks with unreliable links and establishes proofs pertaining to connectivity and k-connectivity. Section 7.5 considers unreliable sensors and establishes a general connection between reliable and unreliable networks. We study some properties of unreliable sensor networks such as connectivity and the existence of the giant component. Section 7.6 contains simulations of these graph theoretic properties, in particular k-connectivity and average path lengths for networks with unreliable links and giant component analysis for networks with unreliable sensors. Finally Section 7.7 concludes the chapter.

7.2 Related Work

Related problems to graph theoretic results in this chapter have been studied in the context of random graph theory [10], continuum percolation and geometric probability [77], [91], [87], [88], [89] and the study of wireless network graphs [44], [42], [125], [11], [12], [115], [32]. In random graph theory, the model $G(n, p)$ is extensively studied, in which edges appear in a graph of n vertices with probability p independently of each other. In continuum percolation theory, usually infinite graphs on $\mathbb{R}^d$ are studied. Finally, in geometric probability and the study of graphs of wireless networks, the graphs in which nodes and links are reliable are

usually studied.

Previously, k-connectivity of wireless networks has been studied in [65] and [124]. In [65] k-connectivity is studied in the context of fault-tolerant networks. The authors find lower bounds for the probability that the network is k-connected. They also present a method to control the network topology given that the network is k-tolerant (k-connected). In [124], authors study the asymptotic critical transmission radius for k-connectivity and asymptotic critical neighbor number for k-connectivity in wireless networks. The connectivity in ad-hoc and hybrid networks is studied in [31]. In [31], authors specifically consider the effect of base stations. They show that the introduction of a sparse network of base stations significantly increases the connectivity. In [30], trade-off between connectivity and capacity of dense networks is studied. In particular, the effect of the attenuation function on network properties is studied. In [29], authors consider a model in which two nodes can communicate if and only if the signal to noise ratio at the receiver is higher than some threshold. Thus, in this way they study the impact of interferences on the connectivity of ad hoc networks.

In this chapter, we consider the connectivity properties of large-scale sensor networks. Thus, we consider the effects of the specific parameters of sensor networks on network properties. In particular, we consider unreliable links, unreliable nodes, and non-uniform distribution of nodes. However, in the papers mentioned above, it is assumed that links and nodes do not experience failures and nodes are distributed uniformly at random over the region. It is sometimes trivial to extend the previous results to include sensor networks (with node and link failures and non-uniform distribution). However, in many cases these new properties of sensor networks introduce new challenges. Thus, in this chapter we need to use new methods for analyzing network properties. It is worth noting that the node failure has been studied in [115]. However, the sensor deployment is confined to a grid and the random distribution of nodes is unexplored. A similar issue to link failures has been studied in [47] in the context of gossip-based routing. They introduce a gossiping-based routing, where each node forwards a message with some probability. However, [47] only provides empirical results. Moreover, in this chapter, we introduce new results about the path lengths and latency in k-connected networks. In particular, we show that multi-path

routing can be done efficiently (in a certain sense) in sensor networks.

In this chapter we also consider multi-path routing with its implicit connection to k-connectivity. Several works have been published in sensor network multiple-path routing. Ganesan et al. [39] introduces multi-path routing in wireless sensor networks and considers disjoint and braided multi-paths. Our work provides the underlying mathematical foundations for which these algorithms may be applied. Multi-path routing in ad hoc networks has also been studied in [81]. Ayanoglu et al. study coding diversity in multiple paths [6]. The results in the following sections, particularly the study of k-connectivity, help to formalize the connectivity and availability of multiple paths in large-scale sensor networks.

Finally, there are some other papers that have empirically studied node failures and the lifetime of wireless sensor networks. Lifetimes of networks have also been considered in terms of energy usage of proposed communications and routing algorithms. Ganesan et al study the presence of patterned and isolated failures as it relates to multi-path routing [39]. In [51], the lifetime of the network is measured in terms of the number of alive nodes as a function of time for a specific routing algorithm in LEACH. There are also comparisons the energy usage over time for several multicast and flooding schemes against proposed algorithm [52, 74]. Other common studies consider the packet delivery ratio [15, 58], but this work considers properties of the network on the link level. This chapter focuses on the broader scope of properties of wireless sensor networks as a whole, including connectivity, average path length, and the presence of a giant component.

7.3 Formulation and Preliminaries

Wireless networks are sometimes modeled by the probability space of graphs that we represent with $g(n, r(n))$. The properties of this model have been studied previously [86], [44], [42]. In this model, it is assumed that n nodes are uniformly and randomly distributed over $S_0 = \overline{S(\overline{O}, 1)}$. If two nodes u and v satisfy $d(u, v) \leq r(n)$ ($d(u, v)$ is the Euclidean distance between u and v), then the edge $\{u, v\}$ belongs to edges of the graph. A more general model is the model $g(n, r(n), f_{XY})$ that is defined as follows. Let X and Y be absolutely continuous random variables with continuous joint density function $f_{XY}(x, y)$

satisfying $f_{XY}(x,y) > 0$ for all $(x,y) \in S_0 = \overline{S(\overline{O},1)}$, and $f_{XY}(x,y) = 0$ otherwise. A graph in $g(n,r,f)$ has n nodes and is generated as follows. For any node v, its position (X,Y) is chosen according to $f_{XY}(x,y)$ independently of other nodes. If two nodes u and v satisfy $d(u,v) \leq r(n)$, then the edge $\{u,v\}$ belongs to edges of the graph.

However, to study sensor networks, we now introduce two new parameters, link failure probability $1-p_e(n)$ and node failure probability $1-p_{sf}(x,y,n) = 1-p_{sf}(x,y)p_{sf}(n)$. We first consider networks experiencing link failures. We introduce the probability space $g(n,r(n),f_{XY}(x,y),p_e(n))$ that we use to model graphs of sensor networks with possibly unreliable links. Let X and Y be absolutely continuous random variables with continuous joint density function $f_{XY}(x,y)$ satisfying $f_{XY}(x,y) > 0$ for all $(x,y) \in S_0 = \overline{S(\overline{O},1)}$, and $f_{XY}(x,y) = 0$ otherwise. A graph in $g(n,r(n),f_{XY}(x,y),p_e(n))$ has n nodes and is generated as follows. For any node v, its position (X,Y) is chosen according to $f_{XY}(x,y)$ independently of other nodes. If two nodes u and v satisfy $d(u,v) \leq r(n)$, then with probability $p_e(n)$ the edge $\{u,v\}$ belongs to the edges of the graph. Note that in the above model sensors are assumed to be reliable. Similar to reliable networks, if $f_{XY}(x,y) = 1_{\{(x,y)\in S_0\}}$ (i.e., nodes are distributed uniformly over the square S_0), we show the corresponding random graph by $g(n,r(n),p_e(n))$.

We then consider node failures. To study sensor networks with unreliable nodes, we define the probability space $g(n,r(n),p_{sf}(x,y,n))$, where $p_{sf}(x,y,n) = p_{sf}(x,y)p_{sf}(n)$. In this model n nodes are uniformly and randomly distributed over S_0; however, a sensor node at the point (x,y) is active with probability $p_{sf}(x,y)p_{sf}(n)$ and fails with probability $1-p_{sf}(x,y)p_{sf}(n)$. The function $p_{sf}(x,y)$ models the possible spatial dependency of failure probability and $p_{sf}(n)$ models possible dependency on n. The nodes that are not active are assumed to be removed from the graph. If two active nodes u and v satisfy $d(u,v) \leq r(n)$, then the edge $\{u,v\}$ belongs to edges of the graph. The generalized model $g(n,r(n),f_{XY}(x,y),p_{sf}(x,y,n))$ is defined similarly. Finally we will consider the combined model $g(n,r(n),f_{XY}(x,y),p_e(n),p_{sf}(x,y,n))$. For simplicity, when there is no danger of confusion, we may drop the arguments, for example we may use $g(n,r,f_{XY})$ instead of $g(n,r(n),f_{XY}(x,y))$. For the purpose of analysis, we divide the square S_0 to different

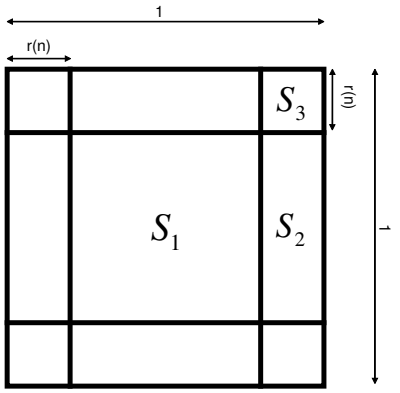

Figure 43: The field S_0 and its divisions S_1,S_2, and S_3.

regions shown in Fig.43.

The following lemma is useful when working on large-scale wireless sensor networks. It can be proved using direct computations and taking limits.

Lemma 21.

$$\lim_{\frac{x}{r}\to o}\left[\frac{\pi r^2 - m\big(\overline{B(\overline{0},r)}\cap\overline{B((0,x),r)}\big)}{2rx}\right]=1. \tag{201}$$

$$\lim_{\frac{x}{r}\to o}\left[\frac{\nu\big(\overline{B((\frac{1}{2}-x,0),r)}\big)-\frac{\pi r^2}{2}}{2rx}\right]=1. \tag{202}$$

We frequently need to find asymptotic behavior of integrals of the form

$$\int_{-\infty}^{+\infty}\varphi(x,n)dx \quad n\to\infty, \tag{203}$$

in which $\varphi(x,n)$ has a sharp peak. These integrals can usually be approximated by the contribution of some neighborhood of the peak. This method is usually called the Laplace method for integrals.

We now quickly review some definitions and results from continuum percolation that we will need later. For a point process χ on $\mathbb{R}^2$ and a Borel set A, let $\chi(A)$ be the number of points of the process in A. The point process is said to be a Poisson process with density $\lambda>0$ if [86]

- For mutually disjoint Borel sets $A_1, A_2, ..., A_k$, the random variables $\chi(A_1), ..., \chi(A_k)$ are mutually independent.

- For any bounded Borel set $A \in \mathcal{B}(\mathbb{R}^2)$ and for every $k \geq 0$, we have

$$\Pr\{\chi(A) = k\} = e^{\lambda m(A)} \frac{\lambda^k (m(A))^k}{k!}. \tag{204}$$

The model for continuum percolation that we use in this chapter is obtained from a Poisson process that is conditioned to have a point at the origin $\chi_\lambda \cup \{\overline{O}\}$ and a connection radius d. In this model two points are connected to each other by an edge if their distance is less than or equal to d. We denote this model by (χ_λ, d) and show the corresponding graph by $g(\chi_\lambda, d)$. Let $p_k(\lambda)$ be the probability that the component of $g(\chi_\lambda, 1)$ containing the origin has k vertices. Then the percolation probability $p_\infty(\lambda)$ is the probability that $\overline{O}$ lies in an infinite component of the graph $g(\chi_\lambda, 1)$, and is defined by [77], [86]

$$p_\infty(\lambda) = 1 - \sum_{k=1}^{\infty} p_k(\lambda). \tag{205}$$

The critical value λ_c which is called the continuum percolation threshold is defined by

$$\lambda_c = \inf\{\lambda > 0 : p_\infty(\lambda) > 0\}. \tag{206}$$

It is well-known that $0 < \lambda_c < \infty$. In particular, we know that $.696 < \lambda_c < 3.372$ [77], [86].

7.4 Networks with unreliable links

We now study the random graph $g(n, r(n), f_{XY}(x, y), p_e(n))$. We first study connectivity (1-connectivity) and then extend the results for general k-connectivity.

7.4.1 Connectivity

We first consider the case where $f_{XY}(x, y) = 1_{\{(x,y) \in S_0\}}$ (i.e., nodes are distributed uniformly over the square S_0). As we discussed, in this case we show the random graph by $g(n, r(n), p_e(n))$. Similar results for general $f_{XY}(x, y)$ will be given later. We first need to prove a lemma. Let $A_{n,1}, A_{n,2}, ..., A_{n,n}$ be a sequence of events in the probability space $(\Omega_n, \mathcal{F}_n, P_n)$. Let $X_{n,j}$ be the random variable defined to be one when $A_{n,j}$ occurs and zero otherwise for $j = 1, 2, ..., n$. Let also $X_n = \sum_{j=1}^{n} X_{n,j}$ be the number of events that occur from the set $\{A_{n,1}, A_{n,2}, ..., A_{n,n}\}$. Define

$$\mu_n = E[X_n] = \sum_{j=1}^{n} \Pr\{A_{n,j}\}, \tag{207}$$

$$\Delta_n = \sum_{i=1}^{n} \sum_{j \neq i} \Pr\{A_{n,i} \cap A_{n,j}\}. \tag{208}$$

We now state the following lemma which is similar to Janson's inequality; however, it is applicable to a more general case.

Lemma 22. *Let $A_{n,j}, \Delta_n, \mu_n$ be as defined. Assume $\lim_{n\to\infty} \mu_n = \mu$, $0 < \mu < \infty$ and $\lim_{n\to\infty} \Delta_n = \Delta$, $0 < \Delta < \infty$. Then*

$$\limsup_{n\to\infty} Pr\{ \bigcap_{i=1}^{n} \overline{A_{n,i}} \} \leq 1 - \mu + \frac{\Delta}{2}. \tag{209}$$

If $\Delta \geq \mu$, then

$$\limsup_{n\to\infty} Pr\{ \bigcap_{i=1}^{n} \overline{A_{n,i}} \} \leq 1 - \frac{\mu^2}{2\Delta}. \tag{210}$$

Proof. We have

$$\begin{aligned} &1 - \Pr\{ \bigcap_{i=1}^{n} \overline{A_{n,i}} \} \geq \\ &\sum_{j=1}^{n} \Pr\{A_{n,j}\} - \sum_{i=1}^{n} \sum_{j<i} \Pr\{A_{n,i} \cap A_{n,j}\} \\ &= \mu_n - \frac{\Delta_n}{2} \end{aligned} \tag{211}$$

Thus,

$$\limsup_{n\to\infty} \Pr\{ \bigcap_{i=1}^{n} \overline{A_{n,i}} \} \leq 1 - \mu + \frac{\Delta}{2}. \tag{212}$$

Now, if $\Delta \geq \mu$, Then $\frac{\mu_n}{\Delta_n} \leq 1 + o(1)$. Let $J \subseteq \{1, 2, ..., n\}$ be chosen in the following way. For any $i \in \{1, 2, ..., n\}$, we have $i \in J$ with probability $\frac{\mu_n}{\Delta_n}(1 - o(1)) \leq 1$ independently. Then, using (211) we have

$$\begin{aligned} &\Pr\{ \bigcap_{i\in J} \overline{A_{n,i}} \} \leq \\ &1 - \sum_{i\in J} \Pr\{A_{n,i}\} + \frac{1}{2} \sum_{i\in J} \sum_{j\in J, j\neq i} \Pr\{A_{n,i} \cap A_{n,j}\}. \end{aligned} \tag{213}$$

By taking expectation, we get

$$E\Big[\Pr\{\bigcap_{i\in J}\overline{A_{n,i}}\}\Big] \leq 1 - E\Big[\sum_{i\in J}\Pr\{A_{n,i}\}\Big] +$$
$$E\Big[\frac{1}{2}\sum_{i\in J}^{n}\sum_{j\in J, j\neq i}\Pr\{A_{n,i}\cap A_{n,j}\}\Big]$$
$$= 1 - \frac{\mu_n}{\Delta_n}\mu_n - \big(\frac{\mu_n}{\Delta_n}\big)^2\frac{\Delta_n}{2} + o(1)$$
$$= 1 - \frac{\mu_n^2}{2\Delta_n} + o(1). \tag{214}$$

In particular, there exists $J \subseteq \{1, 2, ..., n\}$ such that

$$\Pr\{\bigcap_{i\in J}\overline{A_{n,i}}\} \leq 1 - \frac{\mu_n^2}{2\Delta_n} + o(1). \tag{215}$$

Therefore, we obtain

$$\Pr\{\bigcap_{i=1}^{n}\overline{A_{n,i}}\} \leq \Pr\{\bigcap_{i\in J}\overline{A_{n,i}}\} \leq 1 - \frac{\mu_n^2}{2\Delta_n} + o(1). \tag{216}$$

Taking limits we obtain

$$\limsup_{n\to\infty}\Pr\{\bigcap_{i=1}^{n}\overline{A_{n,i}}\} \leq 1 - \frac{\mu^2}{2\Delta}. \tag{217}$$

□

Consider the class of graphs $g(r) = g(n, r, f, p_e)$ in which the radius r is variable and all other parameters are fixed. In other words, to generate a class of graphs from the ensemble, we place n nodes randomly and independently on S_0. For any two nodes v and w, we assign the number x_{vw} which is zero with probability $1 - p_e(n)$ and is 1 with probability $p_e(n)$. Now for a given r, the vertices v and w are connected by an edge if and only if $x_{vw} = 1$ and $d(u, v) \leq r$. Let Q be a property of graphs and let

$$r(Q) = \inf\{r : g(r) \text{ has Q}\}. \tag{218}$$

Let $Q_{c,k}$ be the property of being k-connected and let $Q_{\delta,k}$ be the property that the minimum degree of the graph is at least k. The following result is very similar to the one for the $g(n, p)$ model. It can be shown by using arguments similar to [90] and [89] and we omit the proof due to the space limitation.

Theorem 30. *Given a positive integer k, for almost all $g(r)$ in $g(n, r, f, p_e)$ we have*

$$r(Q_{c,k}) = r(Q_{\delta,k}). \tag{219}$$

We note that if $r(Q_{ce,k})$ is the corresponding threshold for k-edge-connectivity, we have $r(Q_{c,k}) \geq r(Q_{ce,k}) \geq r(Q_{\delta,k})$. Thus Theorem 30 implies that $r(Q_{c,k}) = r(Q_{ce,k}) = r(Q_{\delta,k})$.

Discussion: This theorem states that for large enough networks, the graph is k-connected if and only if the minimum vertex degree is at least k. This is very useful because studying the minimum degree is much simpler than studying k-connectivity. This can also be useful in practice when we want to check the connectivity number of the networks. A simple algorithm is to look at the minimum vertex degree in the graph.

We now consider the connectivity of the random graph $g(n, r, p_e)$. Let $V = \{v_1, v_2, ..., v_n\}$ be the set of vertices of a random graph $g_n = g(n, r, p_e)$ that are uniformly placed on $S_0 = \overline{S(\overline{O}, 1)}$. Suppose $\overline{X}_i = (x_i, y_i)$ is the position of v_i for $i = 1, 2, ...n$ and $B_i = B(\overline{X}_i, r(n))$ is the coverage area of v_i. For any node v_i, if we know the location of the node $\overline{X}_i = (x_i, y_i)$, then the probability that the node is isolated (i.e., the node is not connected to any other node in the graph) is given by

$$\Big(1 - \nu(B_i)p_e(n)\Big)^{n-1}. \tag{220}$$

Since $\overline{X}_i = (x_i, y_i)$ is uniformly distributed over S_0, the probability that a certain node in the graph is isolated is given by

$$n \int_{S_0} \Big(1 - \nu(B(\overline{X}, r(n)))p_e(n)\Big)^{n-1} dm(\overline{X}). \tag{221}$$

Let Z_n be the number of isolated vertices in g_n. Then

$$\begin{aligned} EZ_n &= EZ_n(r(n)) \\ &= n \int_{S_0} \Big(1 - \nu(B(\overline{X}, r(n)))p_e(n)\Big)^{n-1} dm(\overline{X}). \end{aligned} \tag{222}$$

It is easy to prove that EZ_n is a decreasing function of n and there exists $r^*(n)$ satisfying $0 < \lim_{n\to\infty} EZ_n(r^*(n)) < \infty$. We call $r^*(n)$ a threshold of $g_n = g(n, r, p_e)$ for isolated vertices. In fact, as we will see, $r^*(n)$ is a threshold for the property of having isolated vertices in the graph. Thus by Theorem 30, $r^*(n)$ is the connectivity threshold.

Theorem 31. *Let $p_e(n) \geq \frac{c}{\ln n}$, for some constant c. Then $r(n) = r^*(n)$ is a threshold of $g_n = g(n, r, p_e)$ for isolated vertices if and only if*

$$0 < \lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] < \infty. \tag{223}$$

More specifically, $\lim_{n\to\infty} EZ_n(r(n)) = 0$ if and only if $\lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] = \infty$ and $\lim_{n\to\infty} EZ_n(r(n)) = \infty$ if and only if $\lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] = -\infty$.

Discussion: Theorem 31 gives us the threshold for isolated vertices. As we will see asymptotically, this determines the threshold for connectivity. This theorem also reveals an important difference between reliable networks like $g(n, r)$ and unreliable networks such as $g(n, r, p_e)$. To see this, let us examine the condition $p_e(n) \geq \frac{c}{\ln n}$. It is worth noting that the condition $p_e(n) \geq \frac{c}{\ln n}$ is not crucial for our proofs. We can still prove the existence of connectivity thresholds without assuming this condition. However, without this assumption, the results would not have closed form representation. Instead, they would include integrals over the region. Hence, the results would depend on the field shape and boundary. As we will see, by assuming $p_e(n) \geq \frac{c}{\ln n}$, we will obtain very simple conditions for connectivity and the results would not depend on the shape of the sensor field. In fact, although we prove the theorems for S_0, they can be extended to all regions with smooth boundary. Thus unlike reliable networks, in unreliable networks, if $p_e(n)$ is small, the connectivity properties of the networks may depend on the shape of the deployment field. In these networks, unlike the reliable networks, the boundary effects are important. Nevertheless, in most practical applications such as random key distribution schemes, the condition $p_e(n) \geq \frac{c}{\ln n}$ is usually satisfied. This theorem is proved in Appendix D.

The connectivity of $g(n, r, p_e)$ can be characterized by the following theorem.

Theorem 32. *Consider the random graph $g = g(n, r, p_e)$. Let $p_e(n) \geq \frac{c}{\ln n}$, for some constant c. Then g is connected asymptotically almost surely if and only if $\lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] = \infty$.*

Discussion: Theorem 32 gives a necessary and sufficient condition for connectivity of $g(n, r, p_e)$. In particular, we can observe the effect of link failures on the connectivity of the

network. Under the condition $p_e(n) \geq \frac{c}{\ln n}$, the effect of p_e can be modeled by defining an *effective radius* $r_{eff}(n) = \sqrt{p_e(n)}r(n)$. That is, the random graph $g(n, r, p_e)$ is asymptotically almost surely connected if and only if $g(n, r_{eff})$ is connected asymptotically almost surely. However, if the condition $p_e(n) \geq \frac{c}{\ln n}$ does not hold, such an easy interpretation is not possible. The theorem is proved in Appendix D.

Moreover, we can find the distribution of the isolated vertices as follows.

Theorem 33. *Consider the random graph $g = g(n, r, p_e)$ for which $p_e(n) \geq \frac{c}{\ln n}$. Let*

$$r(n) = \sqrt{\frac{\ln n + c}{\pi n p_e(n)}}. \tag{224}$$

Let I_n be the number of isolated vertices in g, which are in $\overline{S(0, 1 - 2r(n))}$. Let $I \in Po(e^{-c})$ (i.e., I has Poisson distribution with mean e^{-c}). Then I_n converges in distribution to I.

Discussion: Theorem 33 gives the distribution of the number of isolated vertices in $g(n, r, p_e)$. First of all, if the condition of Theorem 32 is satisfied, then we should have $c \to \infty$ and thus $e^{-c} \to 0$, which implies that there is no isolated vertices in the network with high probability. This is obviously predictable because the network should be connected in this case. On the other hand, when $c < \infty$ the network is not connected because of some isolated vertices. One way to solve this problem is to increase the communication coverage of the isolated vertices such that they get connected to the rest of the graph. Theorem 33 provides the number of isolated vertices in the network in these situations. Thus, we can estimate the amount of extra transmission power needed for connectivity.

Proof. We use the method of factorial moments to prove the theorem. It suffices to show

$$E(I_n)_k \to e^{-kc} \text{ as } n \to \infty \text{ for } k = 1, 2, ... \tag{225}$$

In fact, for $k = 1$ and 2, this has been shown in the proof of Theorem 32 and it is easily extendable to higher values of k. Let $Is(\overline{X}_1, \overline{X}_2, ..., \overline{X}_k)$ be the probability that the nodes which are located at the locations $\overline{X}_1, \overline{X}_2, ..., \overline{X}_k$ are isolated in $g = g(n, r, p_e)$. Then

$$E(I_n)_k = \tag{226}$$
$$(n)_k \int_{(S_1)^k} Is(\overline{X}_1, \overline{X}_2, ..., \overline{X}_k) dm(\overline{X}_1, \overline{X}_2, ..., \overline{X}_k)$$

Thus, for example by (297) in Appendix D, $E(I_n)_1 \to e^{-c}$ as $n \to \infty$. For $k = 2$, we note that $E(I_n)_2 = (1 - o(1))\Delta_n^1$, where Δ_n^1 defined in (305). Thus, using (306), we conclude that $E(I_n)_2 \to e^{-2c}$ as $n \to \infty$. Finally, we note that this argument can be generalized for an arbitrary k. □

In summary, Theorem 32 gives the necessary and sufficient condition for connectivity for $g = g(n, r, p_e)$. We now generalize this result to any other continuous density function $f_{XY}(x, y)$ as follows. First, note that since S_0 is a compact set in $\mathbb{R}^2$ and $f_{XY}(x, y) > 0$ for all $(x, y) \in \overline{B(0, R)}$, the function $f_{XY}(x, y)$ has a strictly positive minimum on $\overline{B(0, R)}$. We call this minimum f_{min}. The following theorem gives the the necessary and sufficient condition for connectivity of $g = g(n, r, f, p_e)$.

Theorem 34. *Consider the random graph $g = g(n, r, f, p_e)$ for which $p_e(n) \geq \frac{c}{\ln n}$, and $f_{min} = \min\{f_{XY}(x, y), (x, y) \in S_0\}$. Then g is connected asymptotically almost surely if and only if there exists $\omega(n)$ satisfying $\omega(n) \to \infty$ as $n \to \infty$ and $n_0 > 0$ such that*

$$r(n) \geq \sqrt{\frac{\ln n + \omega(n)}{np_e(n)\pi f_{min}}} \quad \text{for } n \geq n_0. \tag{227}$$

Discussion: The main message here is that the connectivity condition is completely determined by the area in the field that has the lowest density f_{min}. Thus, if we have a non-uniform distribution of nodes, assuming the same communication radius, we will need more nodes to obtain a connected network.

Proof. (Sketch) If $r(n) \geq \sqrt{\frac{\ln n + \omega(n)}{np_e(n)\pi f_{min}}}$, then the expected number of isolated vertices in S_0 tends to zero by direct calculation and by comparison with (222). Thus, there are no isolated vertices with high probability. On the other hand, if $\limsup_{n\to\infty} \omega(n)$ in (227) is finite, then for a small enough ϵ, we consider a square S' in S_0 such that $f_{XY}(x, y) < (1 + \epsilon)f_{min}$ for all $(x, y) \in S'$. Then, similar to the proof of Theorem 32, we can show that with a strictly positive probability independent of n, there exists an isolated vertex in S'. □

This theorem implies an interesting property of the uniform distribution:

Corollary 10. *The uniform distribution $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$ requires the lowest amount of transmission power for connectivity.*

If we let $p_e(n)$ be the probability of having a shared secret key between two nodes, then Theorem 34 gives a necessary and sufficient condition for the connectivity of the graph in the general key distribution schemes.

7.4.2 K-Connectivity

In this section we study the k-connectivity property of $g(n, r, f, p_e)$. In summary, the k-connectivity transitions are very sharp. In fact, similar to the situation in $G(n, p)$ model, it can be shown that increasing $\pi r^2(n)p_e(n)\ln n$ by an additive factor $O(\ln\ln n)$ will change the probability of k-connectivity from $o(1)$ to $1 - o(1)$. Although, this can be proved using similar arguments to the previous section, for analyzing sensor networks, we might be interested in a coarser view of the k-connectivity threshold. Again, for simplicity we prove the result for the case $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$, and then state the general result for other densities by considering the minimum value of the density function f_{min}.

Theorem 35. *Consider the random graph $g = g(n, r, p_e)$. Let $p_e(n) \geq \frac{c}{\ln n}$, for some constant c. Assume*

$$\lim_{n\to\infty}\left(\frac{n\pi r^2(n)p_e(n)}{\ln n}\right) = \alpha. \tag{228}$$

Let k be a positive integer. If $\alpha > 1$, then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then g is not k-connected asymptotically almost surely.

Discussion: Note that the condition given here for k-connectivity does not depend on k. We can actually give a more refined condition for k-connectivity, and show that increasing $\pi r^2(n)p_e(n)\ln n$ by an additive factor $O(\ln\ln n)$ will change the probability of k-connectivity from $o(1)$ to $1 - o(1)$. However, in practice the condition given here is sufficient to show the behavior of k-connectivity. An important conclusion that we obtain here is that, the transition from a disconnected graph to a fully k-connected graph is very sharp in large-scale sensor networks. Thus, for example, the graph is actually disconnected with high probability when $\alpha = .99$. On the other hand choosing $\alpha = 1.01$, the graph suddenly becomes k-connected. However, in practice, depending on the network size, we may need to choose a larger α to ensure k-connectivity.

It is also worth noting that for the special case of reliable networks in which $p_e(n) = 1$, the result of Theorem 35 is consistent with [65]. In particular, for reliable networks, it is shown in [65] that if $n\pi r^2(n) \geq \ln n + (2k-1)\ln\ln n - 2\ln(k!) + 2\beta$, then the probability that the network is $(k+1)$-connected is at least $e^{-e^{-\beta}}$. Now for $p_e(n) = 1$, our condition reduces to $\lim_{n\to\infty}\left(\frac{n\pi r^2(n)}{\ln n}\right) > 1$. It easy to see that under this condition we should have $\lim_{n\to\infty}\beta = +\infty$. Thus, the probability of $(k+1)$-connectivity, $e^{-e^{-\beta}}$, converges to one when n approaches infinity, as suggested by Theorem 35. This theorem is proved in Appendix D.

Similar to Theorem 34, we can generalize Theorem 35 to other density functions.

Theorem 36. *Consider the random graph $g = g(n, r, f, p_e)$ for which $p_e(n) \geq \frac{c}{\ln n}$, and $f_{min} = \min\{f_{XY}(x, y), (x, y) \in S_0\}$. Assume*

$$\lim_{n\to\infty}\left(\frac{nf_{min}\pi r^2(n)p_e(n)}{\ln n}\right) = \alpha. \tag{229}$$

Let k be a positive integer. If $\alpha > 1$, then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then g is not k-connected asymptotically almost surely.

As we mentioned previously, k-connectivity is a necessary and sufficient condition for the existence of at least k-disjoint paths between every two vertices in the graph. In sensor networks, we may only need k disjoint paths between the sink and other nodes. However, in large scale sensor networks, this requirement is also equivalent to k-connectivity. The reason is as follows. If the graph is k-connected then obviously there are at least k disjoint paths between the sink and any other node in the graph. On the other hand, if the graph is not k-connected, there is a node in the graph with degree lower than k with high probability by Theorem 30. Thus, there cannot be k disjoint paths between this node and the sink.

7.5 Networks with Unreliable Sensors

7.5.1 Connection Between Reliable and Unreliable Networks

In continuum percolation, unreliable nodes are handled easily by using the Thinning Theorem, which states that an unreliable (with the above definition of reliability) Poisson process is equivalent to a reliable one. For instance, if in the process χ_λ, each node is accepted with probability p and rejected with probability $1-p$, then the resulting process is equivalent to

$\chi_{\lambda p}$, that is a Poisson point process with density λp. However, the relation between reliable graphs ($g(n, r(n), f_{XY})$) and unreliable graphs ($g(n, r(n), f_{XY}, p_{sf})$) is more complicated. In this section, we prove a general result about this relation. This results allows us to find properties of unreliable sensor networks from the well studied model for reliable networks.

Note that a common choice for $p_{sf}(x, y, n)$ is a spatially uniform distribution of unreliability, that is $p_{sf}(x, y, n) = p_{sf}(n)$ for all $(x, y) \in S_0$. However, in some scenarios, sensor nodes at some part of the field may be more prone to failure than other parts. For these situations a spatially non-uniform $p_{sf}(x, y, n)$ is more suitable. We first prove that it suffices to study the uniform $p_{sf}(x, y, n) = p_{sf}(n)$. This is because any $g(n, r(n), f_{XY}(x, y), p_{sf}(x, y, n))$ is equivalent to $g(n, r(n), f'_{XY}(x, y), p'_{sf}(n))$ for some $f'_{XY}(x, y)$ and $p'_{sf}(n)$ as shown in below. Remember we always assume $p_{sf}(x, y, n)$=$p_{sf}(x, y)p_{sf}(n)$.

Lemma 23. *The two models* $g(n, r(n), f_{XY}(x, y),$
$p(x, y, n))$ *and* $g(n, r(n), f'_{XY}(x, y), p'_{sf}(n))$ *are equivalent if*

$$\begin{aligned} f'_{XY}(x, y) &= \frac{f_{XY}(x, y)p_{sf}(x, y)}{\int\limits_{S_0} f_{XY}(x, y)p_{sf}(x, y)dxdy}, \\ p'_{sf}(n) &= \int\limits_{S_0} f_{XY}(x, y)p_{sf}(x, y, n)dxdy. \end{aligned} \tag{230}$$

Discussion: Note that as we mentioned before, in these models we always assume that the failed sensors are removed from the graph. Otherwise, obviously the two models will not be equivalent. The importance of this lemma is its implication that we only need to study $g(n, r(n), f_{XY}(x, y), p_{sf}(n))$. That is, we do not need to consider the dependency of p_{sf} on the location (x, y) because it can be absorbed in $f'_{XY}(x, y)$ as stated in the lemma. This significantly simplifies the analysis.

Proof. Since in both models the location of a sensor nodes and its failure is independent of the other sensor nodes, it suffices to prove that in both models each sensor fails with the same probability, and if it does not fail its location has the same probability distribution in both models. First, we note that in $g(n, r(n), f_{XY}(x, y), p(x, y, n))$ each sensor is active

with probability

$$\int_{S_0} f_{XY}(x,y)p_{sf}(x,y,n)dxdy = p'_{sf}(n), \tag{231}$$

which is the corresponding probability in $g(n,r(n),f'_{XY},p'_{sf}(n))$. Now, if a sensor does not fail, in $g(n,r(n),f'_{XY},p'_{sf}(n))$ its location has the density function $f'_{XY}(x,y)$. In $g(n,r(n),f_{XY}(x,y),p_{sf}(x,y,r$ if a node does not fail its location has the density function

$$\begin{aligned} f'_{XY}(x,y) &= \frac{f_{XY}(x,y)p_{sf}(n)p_{sf}(x,y)}{\int_{S_0} f_{XY}(x,y)p_{sf}(x,y)p_{sf}(n)dxdy} \\ &= \frac{f_{XY}(x,y)p_{sf}(x,y)}{\int_{S_0} f_{XY}(x,y)p_{sf}(x,y)dxdy} \\ &= f'_{XY}(x,y). \end{aligned}$$

□

Thus, from now on we study $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$. We also note that the model $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$ is similar to the $G(\mathcal{P}_\lambda;r)$ defined in [86] in the sense that both have a random number of nodes. However, there is an important distinction between them. The model $G(\mathcal{P}_\lambda;r)$ is simpler to work with because of the spatial independency in the Poisson process. However, we do not have such spatial independency property in $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$. Thus, in [86] the model $G(\mathcal{P}_\lambda;r)$ is used to prove some properties of $g(n,r(n),f_{XY})$ but here we use $g(n,r(n),f_{XY})$ to prove properties of $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$.

Let Q be a property of graphs. Then, $g \in Q$ means the graph g has property Q. The following result establishes a connection between reliable and unreliable networks. It is in some sense similar to the relation between $G(n,p)$ and $G(n,M)$ given in [10], [53] and in fact it is proved using a similar argument. We say that almost every graph in $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$ has Q if $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$ has Q asymptotically almost surely.

Theorem 37. *Let Q be a graph property and let $p_{sf}(n)(1-p_{sf}(n))n \to \infty$ as $n \to \infty$. If for every sequence $m = m(n)$ satisfying $m = np_{sf}(n) + O\left(\sqrt{np_{sf}(n)(1-p_{sf}(n))}\right)$, we have $Pr\{g(m(n),r(n),f_{XY}(x,y))$ has $Q\} \to 1$ as $n \to \infty$, then almost every graph in $g(n,r(n),f_{XY}(x,y),p_{sf}(n))$ has Q.*

Discussion: This is a fundamental theorem that relates unreliable networks to reliable ones. In particular, it shows how to apply any previously known result for reliable networks, to prove the same result for unreliable networks. Note that the theorem is quite general and can be applied to any properties of the networks, not just the connectivity properties.

Proof. Let $N(g)$ be the number of vertices of the graph g and $q(n) = 1 - p_{sf}(n)$. For any positive real number β, let $A_n(\beta)$ be the set of integers m satisfying $|m - p_{sf}(n)n| < \beta\sqrt{p_{sf}(n)q(n)n}$. Let $E_n(\beta)$ be the event that $N(g(n, r(n), f_{XY}(x, y), p_{sf}(n))) \in A_n(\beta)$ and let $E_n^c(\beta)$ be its compliment. Then by Chebyshev's inequality

$$\Pr\{E_n^c(\beta)\} =$$
$$\Pr\{N(g(n, r(n), f_{XY}(x, y), p_{sf}(n))) \in A_n(\beta)\} \leq \frac{1}{\beta^2}.$$

Let also $m_{min}(n)$ be an element of $A_n(\beta)$ with the lowest $\Pr\{g(m(n), r(n), f_{XY}(x, y))\}$ and define m_{max} similarly. Then we have

$$\Pr\{g(n, r, f_{XY}, p)) \text{ has Q}\} \geq$$
$$\Pr\{g(n, r, f_{XY}, p)) \text{ has Q given } E_n(\beta)\}\Pr\{E_n(\beta)\} \geq$$
$$\Pr\{g(m_{min}(n), r(n), f_{XY}(x, y)) \text{ has Q}\}(1 - \frac{1}{\beta^2}) \geq$$
$$(1 - o(1))(1 - \frac{1}{\beta^2}).$$

If we let β tend to infinity, then $1-\frac{1}{\beta^2}$ tends to one, thus we conclude that $\Pr\{g(n, r, f_{XY}, p))$has Q$\}$ is greater than any fixed real number less than one. Thus $\Pr\{g(n, r, f_{XY}, p)$has Q$\}$ tends to one as n goes to infinity. Therefore, almost every graph in $g(n, r, f_{XY}, p)$ has Q. □

Theorem 37 shows how to apply previously proven results for reliable networks to prove the same results for unreliable networks. The converse is also possible for certain properties, although it is less interesting in this chapter. To show the converse we first need some definitions. For two graphs g, g' on $\mathbb{R}^2$, we write $g' \subset_v g$ if g' is obtained by deleting a subset of vertices of g. We say that property Q is increasing if whenever $g' \in Q$ and $g' \subset_v g$ then $g \in Q$. Similarly, we say that property Q is decreasing if whenever $g \in Q$ and $g' \subset_v g$ then $g' \in Q$. Finally Q is said to be convex if $g' \subset_v g \subset_v g''$ and $g', g'' \in Q$ imply that

$g \in Q$. Note that the above definitions are slightly different from the usual definitions of increasing, decreasing, and convex properties in graph theory. Note also that if Q is either increasing or decreasing then it is convex. For example, if Q is the property of having a specific subgraph, then obviously Q is increasing. Therefore, it is also convex.

Suppose Q is an increasing property. Let $p^1_{sf}(n) < p^2_{sf}(n)$ and $p(p^i_{sf}, Q)$ be the probability that $g(n, r, f_{XY}, p^i_{sf})$ has Q for $i = 1, 2$. Using a coupling argument we can easily show that $p(p^1_{sf}, Q) \leq p(p^2_{sf}, Q)$. Thus, if $g(n, r, f_{XY}, p^1_{sf}) \in Q$ with high probability, then $g(n, r, f_{XY}, p^2_{sf}) \in Q$ with high probability, as well. Similarly if Q is decreasing then $p(p^1_{sf}, Q) \geq p(p^2_{sf}, Q)$. Finally if Q is a convex property and we have $g(n, r, f_{XY}, p^1_{sf}) \in Q$ and $g(n, r, f_{XY}, p^2_{sf}) \in Q$ with high probability, then we can conclude for $p^1_{sf}(n) < p^3(n)_{sf} < p^2_{sf}(n)$, $g(n, r, f_{XY}, p^3_{sf}) \in Q$ with high probability. Using this fact, we can prove the following theorem. It states that for convex properties we can use unreliable networks to prove the similar properties for reliable networks. Here, we just state the theorem and omit the proof.

Theorem 38. *Let Q be a convex property and let $p_{sf}(n)(1 - p_{sf}(n))n \to \infty$ as $n \to \infty$. If almost every graph in $g(n, r(n), f_{XY}(x, y), p_{sf}(n))$ has Q, then for fixed real number β, $Pr\{g(m_x(n), r(n), f_{XY}(x, y))$ has $Q\} \to 1$ as $n \to \infty$, where $m_\beta(n) = \lfloor p_{sf}(n)n + \beta\sqrt{p_{sf}(n)q(n)n}\rfloor$.*

Discussion: This is the converse to Theorem 37. In other words, if a result has been previously proven for $g(n, r(n), f_{XY}(x, y), p_{sf}(n))$, for $p_{sf}(n) \neq 0$, we can use this theorem to conclude the same result for $g(n, r(n), f_{XY}(x, y))$.

Finally, we end the section by noting that the number of active nodes has a Gaussian distribution. Let $N(g)$ be the number of (active) vertices of the graph $g = g(n, r(n), f_{XY}(x, y), p_{sf}(n))$ and $q(n) = 1 - p_{sf}(n)$. Define

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt. \tag{232}$$

If $p_{sf}(n)(1 - p_{sf}(n))n \to \infty$, then by the Laplace-Demoivre Theorem we have

$$\begin{aligned} &\Pr\{|N(g(n, r, f_{XY}, p)) - p_{sf}(n)n| < x\sqrt{p_{sf}(n)q(n)n}\} \\ &= (1 + o(1))[\Phi(x) - \Phi(-x)]. \end{aligned} \tag{233}$$

7.5.2 Some Properties of Unreliable Sensor Networks

In this section, we specifically study some important graph theoretic properties of node-unreliable sensor networks. We employ the results in the previous section relating reliable and unreliable networks. These results can be proved directly. However, using the previous work on $g(n, r(n), f_{XY}(x, y))$ and the previous section results, they can be proved in a much simpler way. First, we find the necessary and sufficient condition for k-connectivity. Then, we study another important property in the existence of a giant component. For simplicity we only consider the case that nodes are distributed uniformly over the field. That is $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$. Thus we may use $g(n, r(n), p_{sf}(n))$ and $g(n, r(n))$ to represent $g(n, r(n), f_{XY}(x, y), p_{sf}(n))$ and $g(n, r(n), f_{XY}(x, y))$ respectively.

We now study k-connectivity of $g(n, r(n), f_{XY}(x, y), p_{sf}(n))$. As a special case of Theorem 35 if we let $p_e(n) = 1$, then we obtain the following result.

Corollary 11. *Consider the random graph $g = g(n, r, f_{XY})$ with $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$. Assume*

$$\lim_{n\to\infty} \left(\frac{n\pi r^2(n)}{\ln n} \right) = \alpha. \tag{234}$$

Let k be a positive integer. If $\alpha > 1$, then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then g is not k-connected asymptotically almost surely.

We now prove the following theorem on k-connectivity of unreliable networks.

Theorem 39. *Consider $g = g(n,r(n),f_{XY}(x, y),$*
$p_{sf}(n))$ with $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$ and assume $np_{sf}(n) \to \infty$ as $n \to \infty$. Assume

$$\lim_{n\to\infty} \left(\frac{n\pi p_{sf}(n) r^2(n)}{\ln p_{sf}(n) + \ln n} \right) = \alpha. \tag{235}$$

Let k be a positive integer. If $\alpha > 1$, then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then g is not k-connected asymptotically almost surely.

Discussion: Note that this is very similar to Theorem 35. Thus, one way to prove this, is to use similar proofs given for the previous section. However, as we see applying Theorem 37 makes the proof much simpler.

Proof. We use Theorem 37. Consider a sequence $m = m(n)$ satisfying $m = np_{sf}(n) + O(\sqrt{np_{sf}(n)(1-p_{sf}(n))})$, then

$$\lim_{m(n)\to\infty} \left(\frac{m(n)\pi r^2(n)}{\ln m(n)} \right)$$
$$= \lim_{n\to\infty} \left(\frac{np(1+o(1))\pi r^2(n)}{\ln(np(1+o(1))} \right) = \alpha.$$

Thus, by Theorem 37 and Corollary 11, if $\alpha > 1$, then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, then g is not k-connected asymptotically almost surely. □

So far, we have studied conditions for connectivity of unreliable sensor networks. On the other hand, if a graph is not connected, it can be divided into connected components (disjoint connected subgraphs). In these situations, the sensor network may continue to operate if it has one large component. For the graph $g = g(n, r(n))$ it has been shown in [86] that there exists a threshold $r^*(n)$ such that when $r(n)/r^*(n) < 1$, all components are small (logarithmic in n) with high probability. On the other hand, if $r(n)/r^*(n) > 1$, there exists one giant component (with size linear in n), and other components are small. Note that if the density function is not uniform, there may be more than one giant component. We now generalize these results to unreliable sensors. Again, for simplicity we only consider a uniform distribution of nodes the field, that is $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$. Thus we drop the density function from the notation. The general case of non-uniform distribution can be proved similarly. Let L_j denote the size of the j'th largest component in a graph. We recall that the critical value λ_c is the continuum percolation threshold. The following theorem is proved in [86].

Theorem 40. *Consider the random graph $g(n, r(n))$ and suppose $nr^2(n) \to \lambda$ as $n \to \infty$. Then, if $0 < \lambda < \lambda_c$, there exists a positive constant δ independent of n such that the size of the largest component satisfies $L_1 < \delta \ln n$ with high probability. On the other hand if $\lambda > \lambda_c$, there exists a positive constant α independent of n such that the size of the largest component satisfies $L_1 > \alpha n$ with high probability. Moreover, the size of other components is sublinear. That is, for $j > 1$, $L_j/n \to 0$ as $n \to \infty$ with high probability.*

We now state and prove the corresponding result for unreliable sensor networks, $g(n, r(n), p_{sf}(n))$.

Theorem 41. *Consider the random graph $g(n, r(n), p_{sf}(n))$ and suppose $np_{sf}(n) \to \infty$ and $np_{sf}(n)r^2(n) \to \lambda$ as $n \to \infty$. Then if $0 < \lambda < \lambda_c$, there exists a positive constant δ independent of n such that the size of the largest component satisfies $L_1 < \delta \ln n$. On the other hand if $\lambda > \lambda_c$, there exists a positive constant α independent of n such that the size of the largest component satisfies $L_1 > \alpha np_{sf}(n)$ with high probability. Moreover the size of other components is sublinear. That is for $j > 1$, $L_j/(np_{sf}(n)) \to 0$ as $n \to \infty$ with high probability.*

Discussion: Note that direct proof of this theorem is very involved and cumbersome. However, as we see by using Theorem 37, the proof is almost trivial.

Proof. Again we use Theorem 37. Consider a sequence $m = m(n)$ satisfying $m = np_{sf}(n) + O(\sqrt{np_{sf}(n)(1 - p_{sf}(n))})$, then

$$\begin{aligned}
&\lim_{m(n)\to\infty} m(n)r^2(n) \\
&= \lim_{n\to\infty} \left[np_{sf}(n) + O(\sqrt{np_{sf}(n)(1 - p_{sf}(n))}) \right] r^2(n) \\
&= \lim_{n\to\infty} np(1 + o(1))r^2(n) \\
&= \lambda.
\end{aligned}$$

Thus, if $0 < \lambda < \lambda_c$, by Theorem 40, there exists a positive constant δ independent of n such that the size of the largest component satisfies $L_1 < \delta \ln m(n)$ with high probability. Thus, we conclude that there exists a positive constant δ' independent of n such that $L_1 < \delta' \ln(np_{sf}(n))$. On the other hand if $\lambda > \lambda_c$, there exists a positive constant α independent of n such that the size of the largest component satisfies $L_1 > \alpha m(n)$ with high probability. Thus we conclude that there exists a positive constant α' independent of n such that $L_1 > \alpha' np_{sf}(n)$ with high probability. Moreover, the size of other components is sublinear. Thus, by Theorem 37 we conclude the proof of this theorem. □

7.5.3 Networks with Unreliable Links and Nodes

We can easily combine the results in previous sections to analyze $g(n, r, f, p_e, p_{sf})$. Here we state the results for connectivity.

Corollary 12. *Let Z_n be the the number of isolated vertices in $g_n = g(n, r, p_e, p_{sf})$ and assume $p_e(n) \geq \frac{c}{\ln n}$, for some constant c Then $r(n) = r^*(n)$ is a threshold of g for the existence of isolated vertices if and only if*

$$0 < \lim_{n\to\infty} \left[n\pi r^2(n) p_{sf} p_e(n) - \ln(n)\right] < \infty. \tag{236}$$

More specifically, $\lim_{n\to\infty} EZ_n(r(n)) = 0$ if and only if $\lim_{n\to\infty} \left[n\pi r^2(n) p_{sf}(n) p_e(n) - \ln(n)\right] = \infty$ and $\lim_{n\to\infty} EZ_n(r(n)) = \infty$ if and only if $\lim_{n\to\infty} \left[n\pi r^2(n) p_{sf} p_e(n) - \ln(n)\right] = -\infty$.

Corollary 13. *Consider the random graph $g = g(n, r, f, p_e, p_{sf})$ for which $p_e(n) \geq \frac{c}{\ln n}$, and $f_{min} = \min\{f_{XY}(x, y), (x, y) \in S_0\}$. Then g is connected asymptotically almost surely if and only if there exists $\omega(n)$ satisfying $\omega(n) \to \infty$ as $n \to \infty$ and $n_0 > 0$ such that*

$$r(n) \geq \sqrt{\frac{\ln n + \omega(n)}{n p_{sf}(n) p_e(n) \pi f_{min}}} \quad \text{for } n \geq n_0. \tag{237}$$

Corollary 14. *Consider the random graph $g = g(n, r, f, p_e, p_{sf})$ for which $p_e(n) \geq \frac{c}{\ln n}$, and $f_{min} = \min\{f_{XY}(x, y), (x, y) \in S_0\}$. Assume*

$$\lim_{n\to\infty} \left(\frac{n f_{min} \pi r^2(n) p_{sf}(n) p_e(n)}{\ln n} \right) = \alpha. \tag{238}$$

Let k be a positive integer. If $\alpha > 1$, Then g is k-connected asymptotically almost surely. On the other hand, if $\alpha < 1$, Then g is not k-connected asymptotically almost surely.

7.6 Simulation Results

In this section, we provide some simulations to validate the theoretical development found in the previous sections [1]. The implication of the preceding development is a threshold effect on connectivity under certain network parameters. We show this in networks of varying communication radii ($r(n)$), containing unreliable sensors (sensor failure with probability $1-$

[1] The author wishes to thank Kevin Chan for providing the simulation results in this section.

$p_{sf}(n)$), unreliable links (link failure with probability $1-p_e(n)$) and of varying distributions (f_{min}). Furthermore, the developments have shown that k-connectivity is achieved rapidly at this threshold. We provide the results of simulations to validate these claims.

The results for networks of size $n = \{1000, 2000, 5000\}$ are provided. They have been deployed into a field S_0 of unit dimensions with finite communication radius, $r(n)$. We look at the probability of disconnectivity, p_{disc}, as a function of varying each of the network parameters that we have considered in this work such as $r(n)$, $p_{sf}(n)$, $p_e(n)$, and f_{min}. Looking at the various sizes of networks verifies that the claims are asymptotically valid, as the behavior of connectivity around a threshold is increasingly tighter. This threshold effect is such that for values lower than this threshold, the graph is disconnected with high probability. For values above this threshold, the graph is connected with high probability. Therefore, in the following figures, we represent the theoretical threshold as a step function, where $p_{disc} = 1$ for values below the threshold and $p_{disc} = 0$ past this threshold. The simulations show that the threshold effect occurs situations where each of the network parameters are varied.

We also provide two related characteristics of network connectivity in looking at shortest paths between nodes and the presence of giant components. We look at the relationship between k-connectivity and average shortest path lengths. We show an important result in that the first k shortest paths in a k-connected graph have almost the same length (by the length of a path, we mean the number of hops). Also, we examine the size of the giant component is simulated for networks with unreliable sensor nodes.

7.6.1 Connectivity versus Communications Radius

The threshold for the radius required to provide connectivity has been derived in previous sections. In this section, we provide simulation results to validate the theoretical development of this property (237). As we consider networks of different size, we show that the minimum transmission radius occurs at

$$r(n) \geq (1+\epsilon)\sqrt{\frac{\ln n}{np_{sf}(n)p_e(n)\pi f_{min}}} \tag{239}$$

where ϵ is some small fixed constant. Here we set $\epsilon = .1$.

We assume a fixed, uniform communications radius for each node in the network. Additionally, the nodes are distributed uniformly, $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$, where also $p_{sf}(n) = p_e(n) = 1$. We see that (239) determines the value of $r(n)$ at which this threshold for connectivity should occur. From Figures 44, 45 and 46, we see a threshold effect in p_{disc} as $r(n)$ increases. The effect grows tighter to bound as the size of the network increases. This was expected since the theoretical results are asymptotic and apply to very large networks.

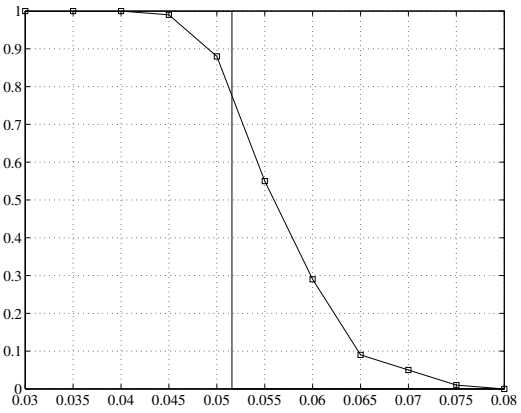

Figure 44: The minimum radius to provide connectivity for a network of size $n = 1000$

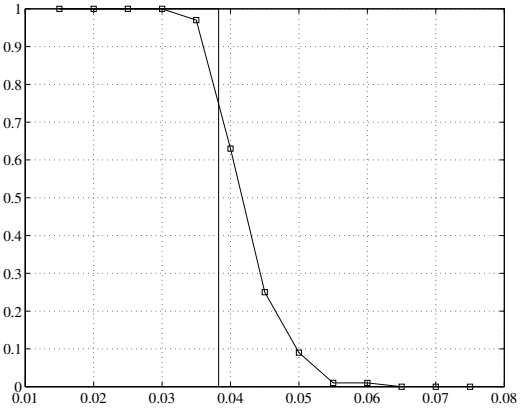

Figure 45: The minimum radius to provide connectivity for a network of size $n = 2000$

7.6.2 Networks with Unreliable Links and Sensors

With (237), we can also derive the requirement for $p_{sf}(n)$ and $p_e(n)$ to achieve connectivity within the network. In this section, we provide simulation results to validate the threshold

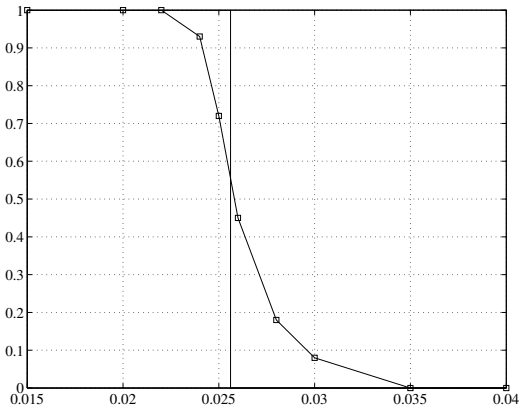

Figure 46: The minimum radius to provide k-connectivity for a network of size $n = 5000$

effect for networks with unreliable links and sensors. We consider the two cases separately, but it also is easy to consider them simultaneously. The sensor failure occurs when after deployment, the node fails to communicate with any device with probability $1 - p_{sf}(n)$. For the link failure, we assume that any link between two nodes within the communication range of each other is formed with probability $p_e(n)$. As we consider networks with failures in either links or sensors, we show that connectivity is achieved at

$$p_{sf}(n) \geq \frac{\ln n}{np_e(n)\pi f_{min} r^2(n)}(1 + \epsilon') \tag{240}$$

and

$$p_e(n) \geq \frac{\ln n}{np_{sf}(n)\pi f_{min} r^2(n)}(1 + \epsilon') \tag{241}$$

In experiment, we assume a fixed, uniform communications radius for each node in the network. Additionally, the nodes are distributed uniformly, $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$, where we have fixed $r(n)$ greater than the threshold of connectivity for $n = \{1000, 2000, 5000\}$, respectively. For instance, for $n = 5000$, we have chosen $r(5000) = .05$, where the threshold value is $r(n) \geq .0256$. Figures 47, 48 and 49 show the probability of disconnectivity versus the values of $p_e(n)$ or p_{sf}. For the plot where we vary $p_e(n)$ we set $p_{sf}(n) = 1$ and where we vary $p_{sf}(n)$ we set $p_e(n) = 1$. We have provided the results for networks of size $n = \{1000, 2000, 5000\}$, respectively. We see a threshold effect in p_{disc} as $p_e(n)$ and $p_{sf}(n)$ increase. The threshold effect is increasingly drastic as the size of the network increases. We note that as the network size increases, the simulation results approach the theoretical threshold.

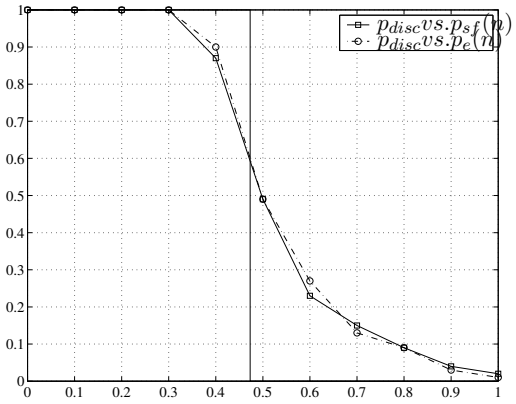

Figure 47: Plot of p_{disc} vs. both p_e and p_{sf} for $n = 1000$.

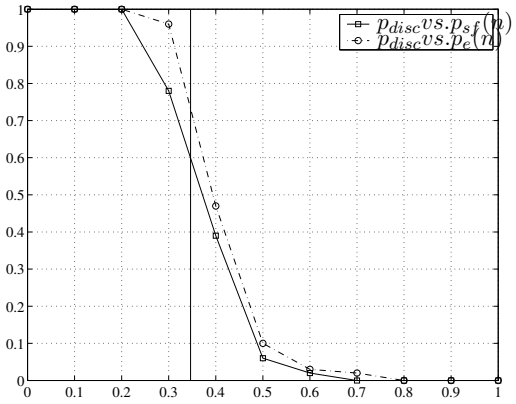

Figure 48: Plot of p_{disc} vs. both p_e and p_{sf} for $n = 2000$.

7.6.3 Connectivity Versus the Distribution of Nodes within Networks

Thus far, we have considered the case where the distribution of the nodes is uniform in S_0. Recall the distribution function $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$. We have also stated that the results are valid for any distribution, where the requirement is dependent on f_{min} the minimum density in S_0.

Therefore, we choose to look at the normal distribution, with truncation. That is, we consider a bivariate normal distribution of nodes on the unit area S_0, only choosing nodes whose coordinates were within S_0. The relationship between σ and f_{min} is determined by (242) and (243).

The distribution of nodes in this case

$$f_{XY}(xy) = \alpha e^{-(x^2+y^2)/2\sigma^2}{}_{\{(x,y)\in S_0\}}. \tag{242}$$

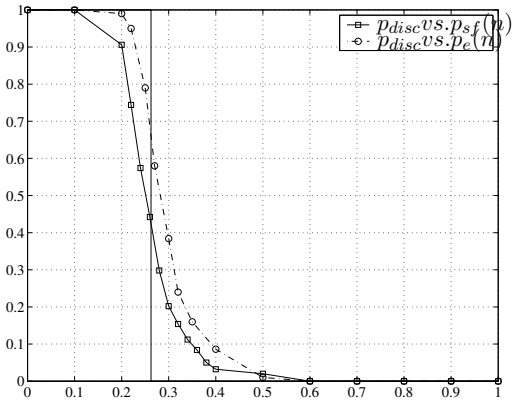

Figure 49: Plot of p_{disc} vs. both p_e and p_{sf} for $n = 5000$.

where

$$\alpha = \left(\int_{-.5}^{.5} \int_{-.5}^{.5} e^{-(x^2+y^2)/2\sigma^2} dydx \right)^{-1} \tag{243}$$

We are able to observe various values of f_{min} by varying the value of σ. In Figure 50, we see that the threshold of p_{disc} and observe that p_{disc} for the truncated bivariate normal distributions follows the general threshold for connectivity. Distributions were generated from several values in $\sigma = [.2, 1]$. Note that we considered reliable sensor networks with $r =$.

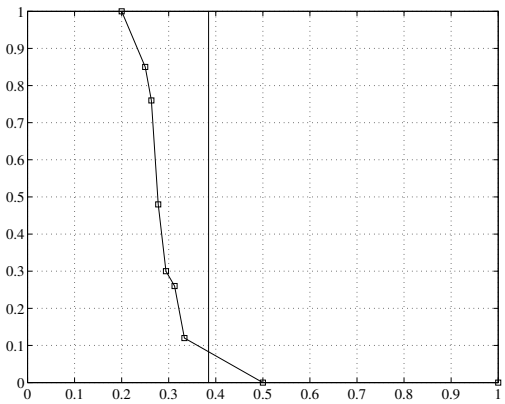

Figure 50: P_{disc} vs. σ for $n = 5000$

7.6.4 Average Shortest Path in k-Connected Networks

In this section we consider k-connectivity. Maintaining a network with several paths when failures in links may occur is important. Furthermore, in routing protocols, multiple paths are used to add redundancy to packet transmission through diversity [6]. Here, we show that

for a k-connected sensor network, the first k shortest paths between two nodes in the network have almost the same length. Therefore, when using multiple paths for transmission, the latency between using different paths does not deviate considerably.

In our simulation, we considered a network of $n = 5000$ nodes with a fixed uniform communication radius of $r(n) = .05$ that ensures k-connectivity for $k < 6$. We also set $p_e(n) = p_{sf}(n) = 1$. We select two nodes in extremal areas of the region S_0. The simulation finds the shortest path between the extremal nodes. Then, the intermediate nodes, those nodes which were used to traverse between the two nodes, are eliminated from the network and the new shortest path is found again. The experiment is repeated to achieve the average shortest path for k-connectivity for $k = \{1, 2, 3, 4, 5\}$. The result of this simulation shows that the average shortest path for $k = \{1, 2, 3, 4, 5\}$ varies by only one hop. This shows confidence that latency among multiple shortest paths does not vary greatly and also demonstrates a great potential for routing algorithms that consider multiple paths. This is a desirable property for algorithms of large-scale sensor networks that employ multiple paths for robust routing and networking schemes. It is also a desirable property for networks with sleeping sensors because it suggests that only a small penalty may be paid if the first shortest path is not used for packet transmission due to sleeping nodes.

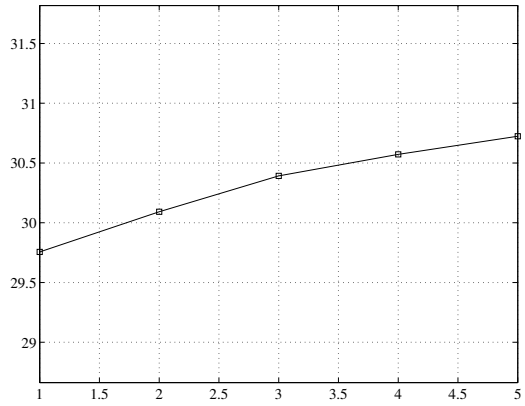

Figure 51: Average Shortest path for $k = \{1, 2, 3, 4, 5\}$, $n = 5000$, $r = .05$,

7.6.5 Giant Component within Networks

In some instances, it may be acceptable to not have full connectivity with all nodes. Instead, a certain proportion of the nodes may be connected and be able to function adequately. In

this simulation, we look to the presence of a giant component within these networks, the largest subset of the active nodes that is connected. We have considered this problem in the case where unreliable nodes exist by considering different values of the number of active nodes in the network by varying $p_{sf}(n)$ in a network of size $n = 10,000$ with communication radius $r(n) = .025$.

In the Theorem 41, the threshold for the network to possess a giant component defined

$$p_{sf} n r^2 = \lambda_c \cong 1.44 \Rightarrow p_{sf} \cong .2304 \tag{244}$$

Therefore the size of the giant component will decrease sharply as $p_{sf}(n)$ decreases below .2304. We have identified this characteristic to be important in the potentially capability of sensor networks. Certainly, it is not desirable for large portions of the network not to be able to communicate with the majority of the nodes. This affects the ability of the nodes to relay information back to the base station.

Figure 52 shows the size of the giant component and the number of active sensors as a function of $p_{sf}(n)$. The solid line represents the average number of active sensors in the network for the specified value of the probability that a node is active, $p_{sf}(n)$, and the dashed lines with boxes is the average size of the giant component in the network. This additionally provides justification of the threshold effect of wireless networks that we have described in this work. The giant component has a threshold effect along with the connectivity.

Collectively, in this section simulation results have verified the theoretical exposition in the preceding sections. We have considered connectivity properties of large-scale networks of varying size. These simulations have confirmed the theoretical developments of unreliable networks with sensor failures and link failures. We have also shown that these claims are valid for other distributions of nodes. Additionally, we have shown that the first shortest paths, on average, are not drastically different in length for the k-connected networks.

7.7 Conclusion

We studied several properties of large-scale sensor networks. We have investigated different graph theoretic properties of sensor networks such as k-connectivity, giant component and

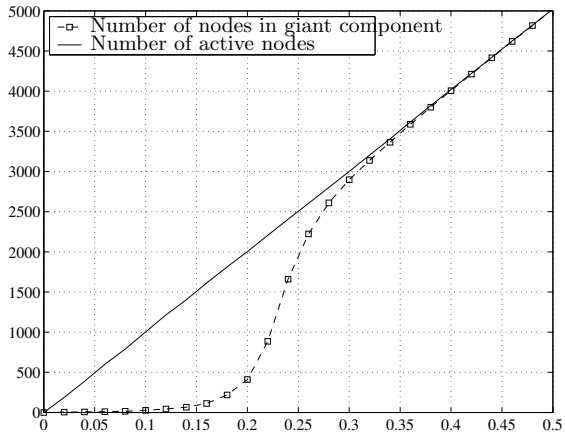

Figure 52: The size of the giant component and the number of active nodes versus $p_{sf}(n)$, the probability that a node is active.

disjoint paths. We considered a model for these networks that includes node and link failures. We proved a general result connecting reliable and unreliable networks. For any positive integer k, we derived the necessary and sufficient conditions for k-connectivity of the sensor network. If $k = 1$, the corresponding condition is the necessary and sufficient condition for connectivity which is clearly an important property of the network. Moreover, k-connectivity is investigated for potential application in multi-path routing or networks with sleeping sensors. The giant component is also studied. We also verified our results by simulation. In particular, we showed that multiple disjoint paths can be found with length very close to the length of the shortest path in a k-connected sensor network. This shows the potential efficiency of multi-path routing in large-scale sensor networks.

CHAPTER VIII

DESIGN AND ANALYSIS OF FINITE WIRELESS NETWORKS

8.1 Introduction

In the past, many analytic results on the connectivity, coverage, and capacity of wireless ad-hoc and sensor networks have been obtained. In almost all of the results, it is assumed that the number of nodes n in the network tends to infinity (large-scale networks). In other words, these results are asymptotic. Asymptotic results are very important for two reasons. First, they give us good estimates for large-scale networks. Second, they show some fundamental trade-offs in the network. However, in many practical wireless networks the number of nodes may be limited to a few hundred (small-scale/finite networks). As it is shown in this chapter, the asymptotic results cease to be valid for these networks. Thus, it is very crucial from practical point of view to analyze finite networks. These analytic results will essentially help us to understand, design, and analyze practical wireless networks, and also to design more suitable communication protocols.

To clarify, let us consider, for example, capacity analysis of wireless networks which has been studied extensively (e.g., in [7, 41–43, 64, 68, 92]). Today we have good understanding of scaling laws in capacity of wireless networks. However, suppose we need to design a wireless sensor network consisting of a hundred sensor nodes. Some fundamental questions are as follows. What is the transport capacity? What are the information theoretic and the MAC layer capacities? How do network parameters such as communication radius of nodes, number of nodes, and so on, affect these capacities? Unfortunately, the available asymptotic results fail to give answers to these questions. Similar questions are remained unanswered for other properties of the network such as connectivity, coverage, etc.

The question that arises here is, can we do small-scale analysis? We recognize some

obstacles as follows. First, in large-scale networks we can use asymptotic estimates that make the analysis much simpler. These estimates are not available in small-scale analysis. Thus, small-scale analysis is usually more difficult. Second, even if we can perform the small-scale analysis, we usually obtain very complicated formulas that are not very useful practically. In this chapter, we want to circumvent these problems and provide guidelines for small scale-analysis. We assume the reader is familiar with large-scale (asymptotic) analyses to some extent. The main goal of the chapter is to initiate the small-scale analysis of wireless sensor and ad hoc networks. Such analyses can be very useful in analyzing and evaluating communication and security protocols for practical sensor and ad hoc networks and is completely overlooked in the literature. To the best of our knowledge, this is the first work to analytically and systematically study this issue.

The main idea is the following. First, for clarity, by small-scale (finite) networks we mean networks of size between $n = 20$ and $n = 2000$, which includes many practical wireless sensor and ad hoc networks. The first key point is to aim at simple and very good approximations instead of trying to find complicated exact formulas. To do so, we first consider the asymptotic analysis. In any asymptotic analysis, a set of asymptotic estimates are used. Some of these estimates are still good for small-scale networks, while others are not. We identify those who are not valid and replace them with better estimates. Specifically, in this chapter we list a few important differences between small-scale and large-scale analysis. Some of these differences, such as the field-shape effect, are specific to random geometric graphs while others apply to all finite and asymptotic systems. Thus, the general method is that we look at any asymptotic analysis and identify the estimations that are not valid for finite networks and replace them with more accurate estimates. However, this must be done carefully, in order to obtain simple and easily computable formulas at the end. As it is mentioned above, exact expressions for network quantities are usually very complicated. Thus, we attempt to provide easily computable estimates for those quantities.

In this chapter, we consider fundamental network properties that affect routing algorithms and reliability of the wireless networks. Specifically, we study coverage, connectivity,

capacity and analysis of routing algorithms. We give several results pertaining to these properties. For example, as an important property we want the network to be connected. More generally we may need k-connectivity. For multi-path routing using k disjoint paths between different nodes, we need the network to be k-connected. Moreover, k-connectivity is related to reliability of networks against node and link failures and adversaries. K-connectivity is also desirable in networks with sleeping sensors. In the past, many authors have studied connectivity and k-connectivity for large-scale networks. These results are asymptotic and obtained assuming that the number of nodes tends to infinity. Here we show that these results are not very useful for finite networks. We provide a very simple formula for k-connectivity probability of wireless networks and show that the formula is very precise.

Related problems have been studied in the context of random graph theory [10], continuum percolation and geometric probability [77, 86], and the study of wireless network graphs [11, 12, 32, 42, 44, 65, 115, 124, 125]. In random graph theory, the model $G(n, p)$ is extensively studied, in which edges appear in a graph of n vertices with probability p independent of each other. In continuum percolation theory, usually infinite graphs on $\mathbb{R}^d$ are studied. Finally, in geometric probability and the study of graphs of wireless networks, large-scale graphs over the plane are usually studied.

In [44], the connectivity of large-scale wireless networks is studied. In [65], [124], and [93], k-connectivity of wireless networks has been studied. In [65], k-connectivity is studied in the context of fault-tolerant networks. In [124] authors study the asymptotic critical transmission radius for k-connectivity and asymptotic critical neighbor number for k-connectivity in wireless networks. In Chapter 7 we studied connectivity and k-connectivity for large-scale sensor networks. We specifically studied the effects of node and link failures and the distribution function of the nodes on connectivity properties of sensor networks. The connectivity in ad-hoc and hybrid networks is studied in [31]. In [30], trade-off between connectivity and capacity of dense networks is studied. In particular, the effect of the attenuation function on network properties is studied. Medium access (MAC) layer capacity of wireless ad hoc networks has been studied in [7]. The transport and information theoretic capacity has been studied extensively, for example see [41–43, 64, 68, 92]. However, almost all previous

analytic results consider graphs in which the number of nodes tend to infinity.

There are also many papers on the empirical study of network characteristics. For example, a survey on routing protocols for wireless sensor networks can be found in [2]. Although many of these papers, consider practical-size networks, they usually rely on simulations. Simulations are a crucial and useful tool for the study of wireless networks; however, as it is discussed in the chapter, they are not enough. Thus, it is very important to have an analytical framework for design and study of wireless networks.

In this chapter, we are concerned with analytical study of finite wireless networks. We show that the practical-size networks need a new analytical treatment. We show that the previous analytic results either do not provide us any results for the finite networks (such as for MAC-layer capacity) or result in very bad estimates. We then introduce a methodology for dealing with finite networks.

The remainder of the chapter is structured into several parts. Section 8.2 establishes the formulation and preliminaries of the problem we have considered. In Section 8.3, we justify the need for small-scale analysis developed in this chapter. In Section 8.4, we investigate the fundamental properties of small-scale analysis. We study coverage, connectivity, capacity, and routing algorithms of finite wireless networks. Finally, Section 8.5 concludes the chapter.

8.2 Preliminaries

We consider a wireless network that consists of n nodes and assume that the nodes are placed on a plane based on a given probability distribution. For example, in wireless sensor networks it is usually assumed that the nodes are randomly and uniformly deployed over a given field [4]. We assume that each node has a finite and fixed communication radius. Two nodes are connected (i.e., can communicate with each other) if they are within communication range of each other. Throughout the chapter, we assume $\mathcal{B}(\mathbb{R}^2)$ is the Borel $\sigma-$algebra on $\mathbb{R}^2$ and m is the Lebesgue measure on $\mathcal{B}(\mathbb{R}^2)$. Note that we just mention measure theoretic definitions to take care of technicalities, and it is not necessary for the reader to be familiar with them. The reader can simply assume that for a set F in $\mathbb{R}^2$, $m(F)$ is the area of F. $\overline{B(\overline{X}, R)}$ is the closed ball with radius R centered at $\overline{X}$ in $\mathbb{R}^2$. $\overline{S(\overline{X}, L)}$ is

the closed square with side L centered at $\overline{X}$ in $\mathbb{R}^2$. In particular $S_0 = \overline{S(\overline{O}, 1)}$ is the closed square with unit area centered at the origin. If u and v are two nodes of a network located in $\mathbb{R}^2$, then $d(u, v)$ is the Euclidean distance between the location of the points. For any set $F \in \mathcal{B}(\mathbb{R}^2)$ we define $\nu(F) = m(F \cap S_0)$. Clearly, ν defines a measure on $\mathcal{B}(\mathbb{R}^2)$. Let ε_n be an event depending on a parameter n. We say that ε_n holds asymptotically almost surely if $\Pr\{\varepsilon_n\}$ tends to 1 as $n \to \infty$.

Wireless networks are sometimes modeled with the probability space of graphs that we represent with $g(n, r) = g(n, r(n))$. In this model, it is assumed that n nodes are uniformly and randomly distributed over $S_0 = \overline{S(\overline{O}, 1)}$. If two nodes u and v satisfy $d(u, v) \leq r(n)$, then the edge $\{u, v\}$ belongs to edges of the graph. A more general model is the model $g(n, r, p)$, in which two nodes are connected with probability $0 < p \leqslant 1$ if their distance is less than r. In this model p models link failures that are common in wireless networks. Asymptotic properties of $g(n, r)$ have been studied extensively. Here we are interested in these properties when n is not necessarily large. It is worth noting that the assumption that the nodes are distributed on S_0 is made for simplicity. These arguments can easily be generalized to other models for the deployment region.

Another generalization is given by $g(n, r(n), f_{XY})$, which is defined as follows. Let X and Y be absolutely continuous random variables with continuous joint density function $f_{XY}(x, y)$ satisfying $f_{XY}(x, y) > 0$ for all $(x, y) \in S_0 = \overline{S(\overline{O}, 1)}$, and $f_{XY}(x, y) = 0$ otherwise. A graph in $g(n, r, f)$ has n nodes and is generated as follows. For any node v, its location (X, Y) is chosen according to $f_{XY}(x, y)$ independently from other nodes. If two nodes u and v satisfy $d(u, v) \leq r(n)$, then the edge $\{u, v\}$ belongs to edges of the graph. Here for simplicity, we restrict ourselves to the model $g(n, r(n))$ and $g(n, r, p)$ (i.e., when $f_{XY}(x, y) = 1_{\{(x,y)\in S_0\}}$, the uniform distribution of nodes). Again, all the arguments can be easily extended to a general density function $f_{XY}(x, y)$. For the purpose of analysis, we divide the square S_0 to different parts shown in Fig.43.

Finally, we consider the following definition for Poisson processes. For a point process χ on $\mathbb{R}^2$ and a Borel set A, let $\chi(A)$ be the number of points of the process in A. The point process χ_λ is said to be a Poisson process with density $\lambda > 0$ if [86]

- For mutually disjoint Borel sets $A_1, A_2, ..., A_k$, the random variables $\chi(A_1), ..., \chi(A_k)$ are mutually independent.

- For any bounded Borel set $A \in \mathcal{B}(\mathbb{R}^2)$ and for every $k \geq 0$, we have

$$\Pr\{\chi(A) = k\} = e^{-\lambda m(A)} \frac{\lambda^k (m(A))^k}{k!}. \tag{245}$$

8.3 Motivation for Small-Scale Analysis

In this section, we present some evidence to show that previous asymptotic results diverge significantly from actual values for finite networks. To show this, we consider connectivity. We first provide the asymptotic probability of disconnectivity for $g(n, r, p)$ and compare it to simulation results. Using this, we conclude that the asymptotic results fail to provide an acceptable estimate of real probability of disconnectivity for small-scale networks. The following result is proved in [44], where a slightly different model is considered. However, the results can be trivially extended to $g(n, r)$ as:

Theorem 42. *(Gupta and Kumar 1998) Let $c_n = n\pi r^2 - \log(n)$, then $g(n, r)$ is connected with high probability if $\lim_{n\to\infty} c_n = \infty$. On the other hand, if $\lim_{n\to\infty} c_n = c < \infty$ then for large n, $g(n, r)$ is disconnected with a strictly positive probability $1 - p_{asymp}(c)$.*

This theorem states that if $\lim_{n\to\infty} c_n = c < \infty$, the network connectivity probability will be bounded away from one. In fact, $p_{asymp}(c)$ is the limit for the probability that the network is connected when n goes to infinity. Although, there is vast literature on the asymptotic analysis of connectivity properties of wireless networks, we were unable to find a reference that actually gives a formula for $p_{asymp}(c)$. Thus, here we compute $p_{asymp}(c)$. Since, the asymptotic study is not the purpose of this chapter, we just provide the summary of the proof.

Theorem 43. *Let $c_n = np\pi r^2 - \log(n)$, and $\lim_{n\to\infty} c_n = c < \infty$, then the probability that $g(n, r, p)$ is connected , $p_{asymp}(c)$, satisfies*

$$p_{asymp}(c) = \lim_{n\to\infty} p_{asymp}(c, n) = e^{-e^{-c}}. \tag{246}$$

Proof. (sketch) We first note that with high probability $g(n,r)$ is connected if there is no isolated vertices [86]. Second, we prove that with high probability there is no isolated vertices in S_2 and S_3 in Fig. 43. In particular, Let $Y_{3,n}$ be the number of isolated vertices in S_3. Then

$$\begin{aligned} EY_{3,n} &\leq nr^2(n)\left(1-\frac{\pi r^2(n)}{4}p\right)^{n-1} \\ &\leq nr^2(n)e^{-\frac{\pi r^2(n)}{4}p(n-1)}. \end{aligned} \tag{247}$$

Using $\pi r^2(n) = \frac{\log n + c_n}{n}$, we conclude

$$EY_{3,n} = O(\frac{\log n(\log n + c)e^{-c/4}}{n^{\frac{1}{4}}}) = o(1). \tag{248}$$

Therefore, there is no isolated vertex in S_3 with high probability. Next, let $Y_{2,n}$ be the number of isolated vertices in S_2. Then

$$EY_{2,n} = n\int\limits_{S_2}\left(1-\nu(B(\overline{X},r(n)))\right)^{n-1}dm(\overline{X}). \tag{249}$$

Using the Laplace method for integrals it can be shown that

$$EY_{2,n} = O(\frac{e^{-\frac{c}{2}}}{r(n)\sqrt{n}}) = o(1). \tag{250}$$

Thus $Y_{2,n} = 0$ asymptotically almost surely. It remains to study the isolated vertices in S_1. Let I_n be the number of isolated vertices in S_1. Using the method of moments we can prove the following. Let $I \in Po(e^{-c})$ (i.e., I has Poisson distribution with mean e^{-c}). Then I_n converges in distribution to I. Thus we conclude that $p_{asymp} = e^{-e^{-c}} = \lim\limits_{n\to\infty} e^{-ne^{-n\pi r^2}}$. □

Therefore, asymptotically, the probability that $g(n,r,p)$ is connected is given by $p_{asymp} = e^{-ne^{-np\pi r^2}}$. We now show that the above asymptotic connectivity formula results in a very bad estimate of disconnectivity probability for small-scale networks. However, in the next sections, we will confirm that our small-scale analysis gives a very good estimate for this quantity.

In Figure 53, we compare the probability of having a disconnected graph for $n = 100$ and $p = 1$ derived by exhaustive simulations and the asymptotic result. In the figure, the

probability of disconnectivity is shown as a function of r, the communication radius. The experiment shows that these results may differ by 10 orders of magnitude. This illustrates that the asymptotic method fails to provide a good approximation for small- scale networks.

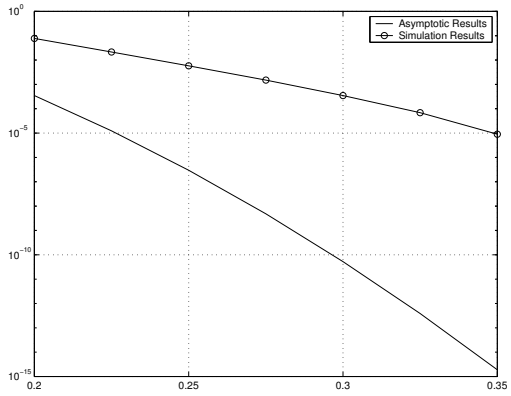

Figure 53: Comparison of asymptotic results with the small scale simulation results for the probability of disconnectivity of $g(n = 100, r)$.

So far we showed that the asymptotic analysis may fail badly for finite networks. Now if we ought to use finite-scale analysis, what kind of formulation would be helpful? To answer, let us now elaborate on the important requirement we mentioned earlier. Namely, in small-scale analysis we need to find simple and easily computable formulas. The rationale behind this is as follows. First, in analytic results we usually need formulas that help us to understand the effects of different parameters. A complicated formula usually reveals little about those effects. Second, sometimes, exact formulas are computationally infeasible. To show this, let us again consider connectivity.

For n points $\overline{X}_1, \overline{X}_2, ..., \overline{X}_n$ on the plane, let the graph $g(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n, r)$ be obtained as follows. The graph consists of n nodes $v_1, v_2, ..., v_n$, such that v_i is located at $\overline{X}_i$. Two nodes v_i and v_j are connected by an edge if their distance from each other is less than r. Let $Con(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n, r) = 1$ if the graph $g(X_1, X_2, ..., X_n, r)$ is connected and $Con(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n, r) = 0$ otherwise. Then the probability that $g(n, r)$ is connected is exactly given by

$$\int_{(S_0)^n} Con(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n) dm(\overline{X}_1, \overline{X}_2, ..., \overline{X}_n).$$

Although, it may not be very obvious, this formula is computationally infeasible. Thus, unless for very small values of n, such as $n = 2, 3$, it is practically useless. Moreover, this formula does not reveal anything about the interplay between different network parameters such as r and n and network properties. This example shows that obtaining exact formulas is not usually enough. Instead, we need to find meaningful and easily computable formulas.

Finally, we note that for some network quantities such as connectivity probability, it is possible to perform exhaustive simulations to estimate the quantity. Nevertheless, it is still very important to analytically study the network properties. First, analytic study helps us to understand the network behavior and see the effects of different parameters on the network properties. Thus, analytic results are very valuable in the design and evaluation of wireless networks. Second, there are many other network quantities that may not be evaluated by exhaustive simulations. For example, in this chapter, we analytically study the capacity of wireless networks. It is unclear, if it is possible to set up simulations to estimate the network capacity [1]. Third, quantities such as connectivity probability are usually used in the analysis of more complicated network properties such as capacity analysis. Thus, it is important to analytically study them. Here is a simple analogy. In circuit design, we can always use the specialized computer packages to analyze a circuit. However, it is still very important to understand the behavior of different components of a circuit. A circuit designer must have access to analytic formulas and basic understanding of the circuit design methodology to design a circuit. Later computer simulations, can be helpful in validating the design, obtaining more exact evaluations, and making final adjustments.

8.4 Fundamentals of Small-Scale Analysis

In this section, we try to establish a framework for analysis of finite networks. We list some important differences between small-scale and large-scale networks. In each subsection we first introduce the main idea, and then pick one or two network properties and show how

[1] Note that we can estimate the average throughput for a given network with a specific protocol and data traffic model using exhaustive simulations. However, here, by capacity we mean the highest possible achievable capacity, not the one achieved using a specific communication and routing protocol. Such capacity measure can be used to determine the efficiency of different protocols.

to analyze those properties for small-scale networks. We try to choose simple examples that best show the difference between small-scale and large-scale analysis. In most cases, for simplicity, we only consider $g(n, r)$; while occasionally we give the results for the more general model $g(n, r, p)$. Nevertheless, it is not usually very difficult to extend the given results for $g(n, r)$, to $g(n, r, p)$.

8.4.1 Boundary Effects

One important phenomenon in asymptotic analysis is that boundary effects can be neglected. Loosely speaking, the analysis of the network properties is usually dominated by what happens in region S_1 in Figure 43. In fact, we saw an example of this phenomenon in the asymptotic analysis of connectivity in Theorem 43. This can considerably simplify the analysis and results in simple and closed-form formulas for network properties. However, in small-scale networks boundary effects cannot be neglected. In other words, nodes in the corners of the field can play an important role in some of the network properties. To clarify this, let us consider a simple example. Suppose we want to find the average coverage in a wireless sensor network defined by $g(n, r)$. In other words, we want to find the average percentage area that is covered. For simplicity, suppose the sensing radius is also equal to r, that is, each node covers a circle of radius r centered at the node location. The probability that the point $\overline{X}$ in S_0 in not covered is given by

$$\Big(1 - \nu(B(\overline{X}, r))\Big)^n. \tag{251}$$

Thus, if PC_{notcov} is the average percentage of the uncovered area, then

$$PC_{notcov} = \int_{S_0} \Big(1 - \nu(B(\overline{X}, r))\Big)^n dm(\overline{X}). \tag{252}$$

Thus, PC_{notcov} can be obtained easily and (252) is valid for all values of n. However, in asymptotic analysis, assuming that $\lim_{n\to\infty} r(n) = 0$ (this assumption is almost always true

for the large-scale analysis), and using (252), we obtain

$$PC_{notcov} = m(S_1)\left(1-\pi r^2\right)^n + \int\limits_{S_0 \setminus S_1} \left(1-\nu(B(\overline{X},r))\right)^n dm(\overline{X})$$
$$= (1-o(1))(1-\pi r^2)^n. \tag{253}$$

Therefore, asymptotically, $PC_{notcov} = (1-\pi r^2)^n$. Note that in this case the only difference between the exact (formula (252)) and asymptotic expressions comes from the edge effect. Figure 54 compares the two results. We observe that the two results differ considerably. This example clearly shows the importance of boundary effects in small-scale networks. This example is unique in the sense that the exact analysis is very simple. However, this is not often the case. For instance, as we will see, exact analysis of other properties can be very complicated.

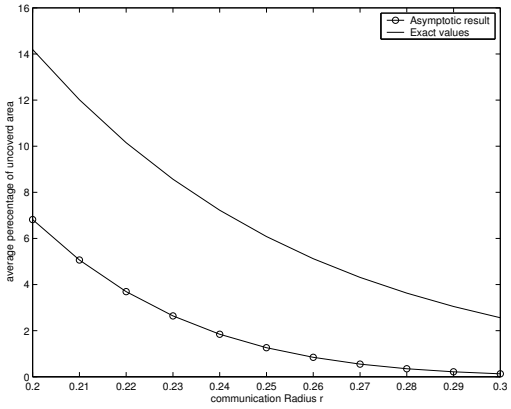

Figure 54: Comparison of asymptotic results with the exact values for average percentage of uncovered area in $g(20,r)$.

Small-Scale Analysis for Connectivity Properties of g(n,r,p):

Before discussing other differences between large-scale and small-scale analysis, we provide a small-scale analysis for connectivity properties of $g(n,r,p)$. This is a good example to illustrate our methodology for small-scale analysis. Since the exact analysis is usually very difficult or at least results in very complicated formulas, a good approach is to find simple lower and upper bounds. Therefore, in this section we find lower and upper bounds for the probability that $g(n,r,p)$ is disconnected, $p_{disc}(n,r,p)$. As we will see the two bounds

almost coincide with each other. Thus, they give a very good estimate for $p_{disc}(n,r,p)$. Indeed, the two bounds completely agree with the simulation results. Let $p_{low}(n,r,p)$ and $p_{upp}(n,r,p)$ be the lower and upper bounds on $p_{disc}(n,r,p)$, respectively. Here we consider the case where $p_{disc}(n,r,p)$ is small (i.e., $p_{disc}(n,r,p) < .1$). In practice, this is usually the range that is important, since we want a network that is connected with high enough probability.

Theorem 44. *Consider a wireless network with the model $g(n,r,p)$. Then we have*

$$\begin{aligned} p_{disc}(n,r,p) \ &\geq n \int\limits_{S_0} \Big(1-\nu(B(\overline{X},r))p\Big)^{n-1} dm(\overline{X}) - \\ &\binom{n}{2} \int\limits_{S_0}\int\limits_{S_0} \Big(1-\nu(B(\overline{X},r))p - \nu(B(\overline{X},r))p + \\ &\nu(B(\overline{X},r) \ \cap B(\overline{Y},r))p^2\Big)^{n-2} dm(\overline{X}) \times m(\overline{Y}). \end{aligned} \tag{254}$$

Proof. Let $p_1(n,r,p)$ be the probability that there exists at least one isolated node (a vertex with no neighbors) in $g(n,r,p)$. Let also $v_1, v_2, ..., v_n$ be the n vertices of $g(n,r,p)$. Then $p_{disc}(n,r,p) \geq p_1(n,r,p)$. Applying the inclusion-exclusion lemma we obtain

$$\begin{aligned} p_{disc}(n,r,p) &\geq \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} \Pr\{v_1, v_2, ..., v_k \text{ are isolated vertices}\} \\ &\geq n \Pr\{v_1 \text{is isolated }\} - \binom{n}{2} \Pr\{v_1 \text{and} v_2 \text{ are isolated vertices}\}. \end{aligned} \tag{255}$$

Note that

$$\Pr\{v_1 \text{is isolated }\} = \int\limits_{S_0} \Big(1-\nu(B(\overline{X},r))p\Big)^{n-1} dm(\overline{X}). \tag{256}$$

Now define $Circ(a,b,r) = \{(x,y) : (x-a)^2 + (y-b)^2 \leq r^2\}$. Then we have

$$\begin{aligned}
&\Pr\{v_1 \text{and} v_2 \text{ are isolated vertices}\} = \\
&\int_{S_0} \int_{S_0 \setminus Circ(\overline{X},r)} \Big(1 - \nu(B(\overline{X},r))p - \nu(B(\overline{X},r))p + \\
&\nu(B(\overline{X},r) \cap B(\overline{Y},r))p^2\Big)^{n-2} dm(\overline{X}) \times m(\overline{Y}) + \\
&(1-p) \int_{S_0} \int_{Circ(\overline{X},r)} \Big(1 - \nu(B(\overline{X},r))p - \nu(B(\overline{X},r))p + \\
&\nu(B(\overline{X},r) \cap B(\overline{Y},r))p^2\Big)^{n-2} dm(\overline{X}) \times m(\overline{Y}) \leq \\
&\int_{S_0} \int_{S_0} \Big(1 - \nu(B(\overline{X},r))p - \nu(B(\overline{X},r))p + \\
&\nu(B(\overline{X},r) \cap B(\overline{Y},r))p^2\Big)^{n-2} dm(\overline{X}) \times m(\overline{Y}).
\end{aligned} \tag{257}$$

Combining (255)(256), and (257) we conclude the theorem. □

Note that this lower bound for $p_{disc}(n,r,p)$ may seem to be too complicated and thus may not satisfy the simplicity requirement. However, as we will see, this lower bound is almost the same as a simple upper bound that we find shortly. Thus, the simple upper bound can be used in estimating $p_{disc}(n,r,p)$. The lower bound is useful in the sense that it assures us that our estimate is very close to the real value for $p_{disc}(n,r,p)$.

We now find an upper bound for $p_{disc}(n,r,p)$. By definition, a connected component of a graph g is a connected subgraph that is isolated from the rest of g. Thus, $p_{disc}(n,r,p)$ is equal to the probability that $g(n,r,p)$ has at least one component of size less than $n/2$. For $U \subseteq \{v_1, v_2, ..., v_n\}$, let $p_{comp}(U)$ be the probability that the vertices in U construct a connected component in $g(n,r,p)$. Then, we have

$$p_{disc}(n,r,p) \leq \sum_{k=1}^{n/2} \binom{n}{k} p_{comp}(\{v_1, v_2, ..., v_k\}) = \sum_{k=1}^{n/2} a_k. \tag{258}$$

Note that although $p_{upp}(n,r,p) = \sum_{k=1}^{n/2} a_k$ is a valid upper bound for $p_{disc}(n,r,p)$, it does not satisfy the simplicity requirement. In fact, except the first few terms, computing a_k's is computationally infeasible. We now try to simplify this upper bound. Note that so

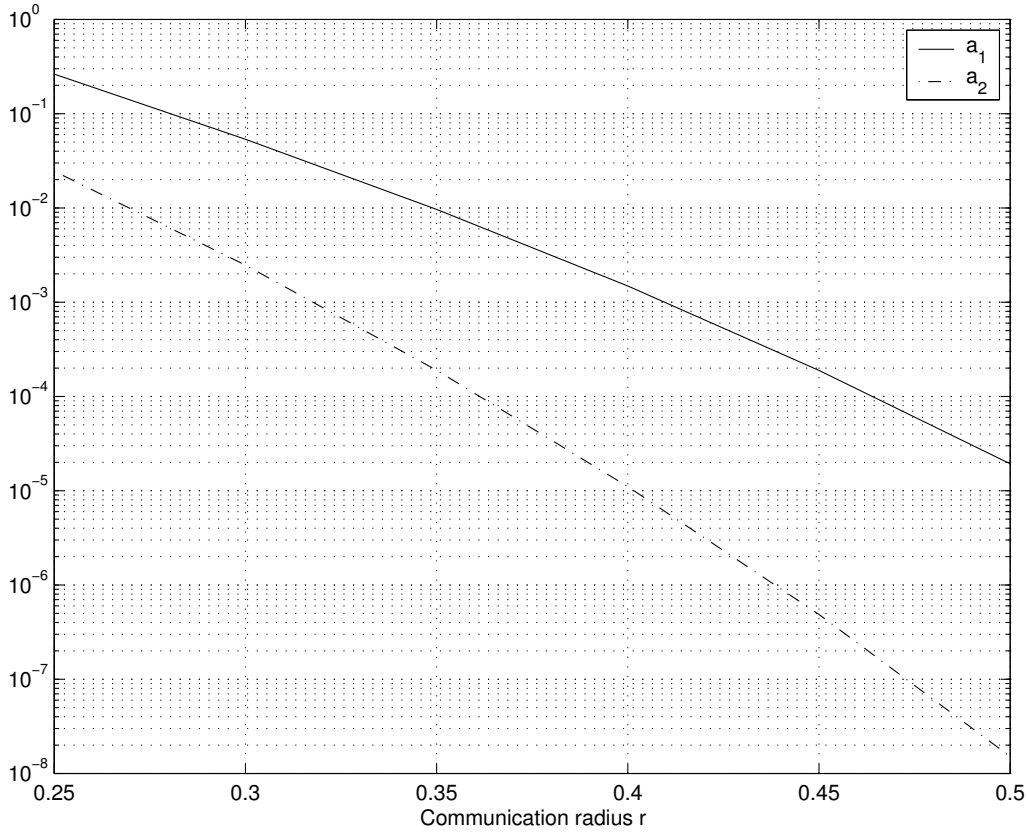

Figure 55: Comparison of a_1 and a_2 in (258).

far all the results concerning the lower and upper bounds have been exact and rigorous. However, we now use a simple approximation for simplifying the upper bound. Nevertheless, the approximation is completely backed by numerical and analytical arguments. We remember our assumption that $p_{disc}(n, r, p)$ is not very large, specifically we assume $p_{disc}(n, r, p) < .1$. An important observation here is that, by this assumption, a_k's decay very fast and the upper bound $p_{upp}(n, r, p) = \sum_{k=1}^{n/2} a_k$, is dominated by a_1. This can be seen by both numerical simulations and intuitive analytical arguments. To see this let us examine a_1 and a_2. Figure 55 compares a_1 and a_2 for $g(n = 100, r, p = .5)$. As we see a_2 is at least one order of magnitude lower than a_1. Note that this is a crucial observation that simplifies the upper bound significantly.

The fact that a_k's decay very fast, can also be described in the following way. For a subset of vertices $U = \{u_1, u_2, ..., u_t\} \subseteq \{v_1, v_2, ..., v_n\}$, let $A(U)$ be the area of the unions of circles with radii r centered at u_i's. Then the probability that the vertices in U are isolated

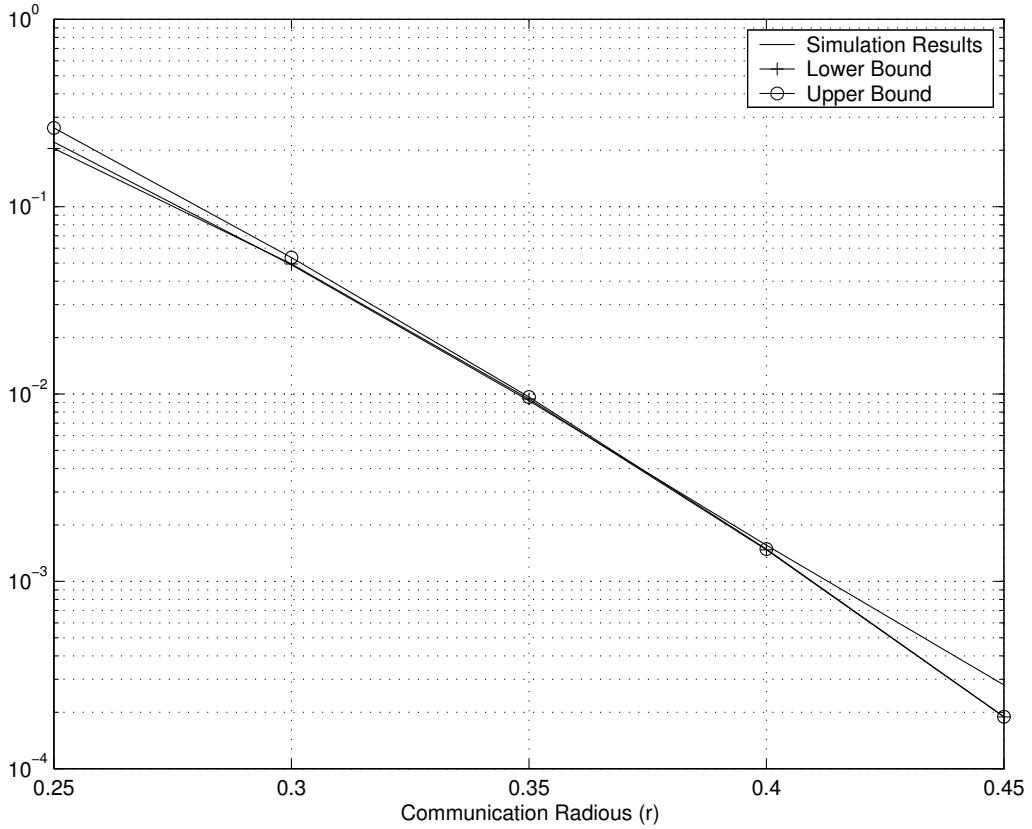

Figure 56: Disconnectivity probability of $g(100, r, .5)$: lower bound, upper bound, and the simulation results.

from the rest of the graph is given by

$$\big(1 - A(u)\big)^{n-t} \simeq e^{-nA(u)}. \tag{259}$$

This shows that $p_{comp}(\{v_1, v_2, ..., v_k\})$ in 258, decays exponentially fast with the number of vertices, k. Thus, a_k's decay very fast. This is of course consistent with our observation in Fig. 55. Therefore, we conclude

$$p_{upp}(n, r, p) \simeq a_1 = n \int\limits_{S_0} \Big(1 - \nu(B(\overline{X}, r))p\Big)^{n-1} dm(\overline{X}). \tag{260}$$

Figure 56 shows the upper bound, lower bound, and the simulation result for the probability of disconnectivity of $g(n, r, p)$, for $n = 100$, and $p = .5$. As we see the three almost coincide. As we will see shortly, similar results are achieved if we use different choices of parameters. Thus, we conclude

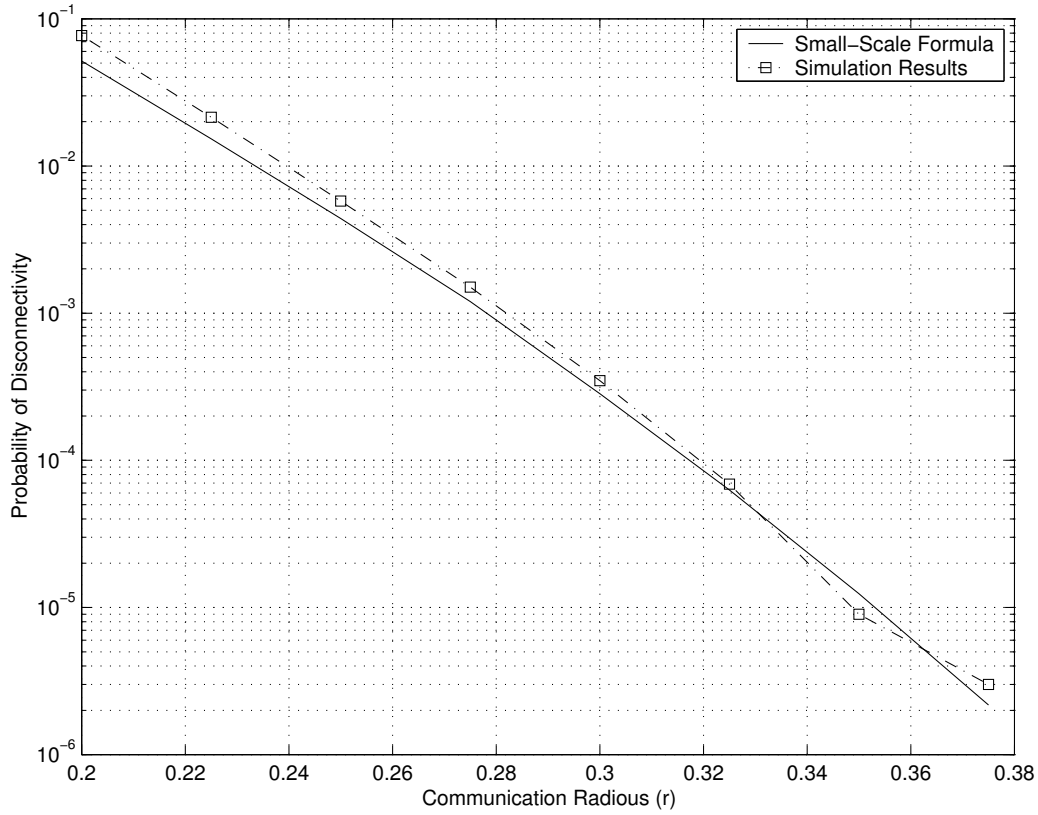

Figure 57: Disconnectivity probability of $g(100, r, 1)$ using (261) and simulation results.

$$p_{disc}(n, r, p) \simeq n \int_{S_0} \Big(1 - \nu(B(\overline{X}, r))p\Big)^{n-1} dm(\overline{X}). \tag{261}$$

Note that (261) suggests that $p_{disc}(n, r, p)$ is dominated by the probability of having an isolated vertex. We recall from Theorem 43 that the asymptotic probability of disconnectivity, $1 - p_{asymp}(c)$, is also dominated by the isolated vertices. However, a crucial difference between these two is that the boundary effects are insignificant in asymptotic analysis. This causes that the asymptotic formula differs from the correct values by several orders of magnitude when used for small or moderate values of n as shown in Fig. 53. However, our small-scale formula is almost identical to the correct values because it considers the boundary effects. Note that (261) gives us a very simple and easily computable formula for disconnectivity probability.

Figures 57, 58, and 59 compares the disconnectivity probabilities obtained by (261) and simulations for different values of n and p. We confirm that in all the cases the given formula

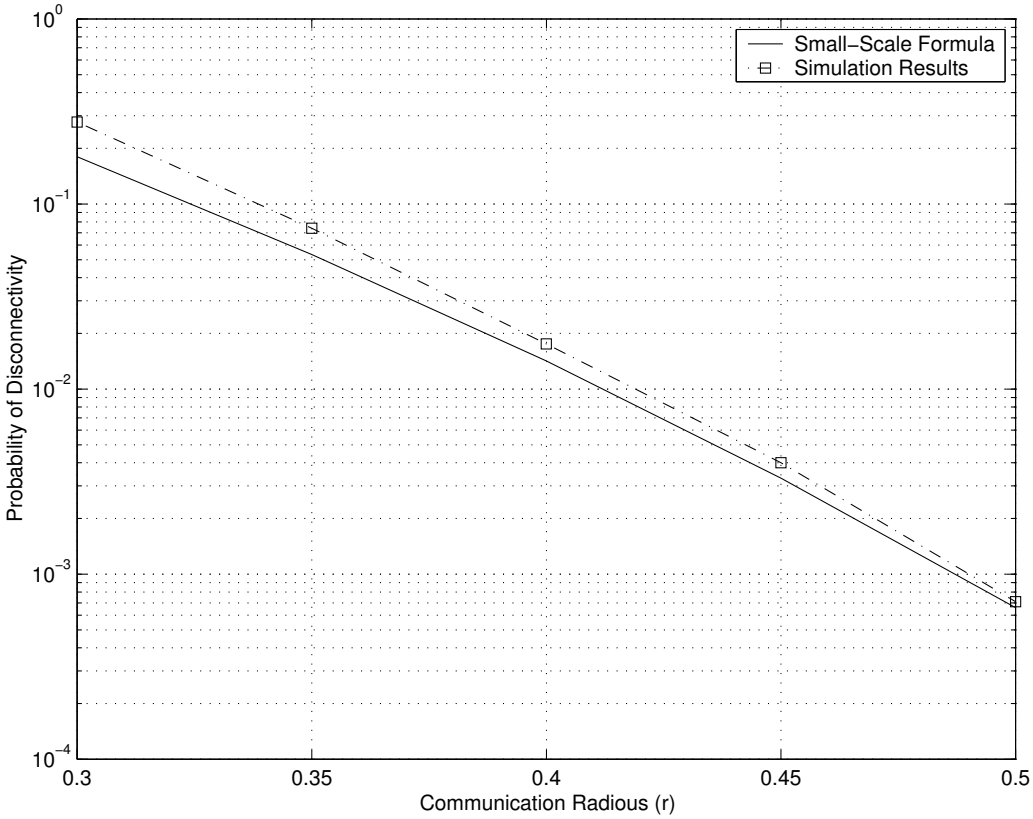

Figure 58: Disconnectivity probability of $g(30, r, 1)$ using (261) and simulation results.

matches the simulation results.

It is worth noting that the methodology used here can be used to study k-connectivity which is more general than connectivity. As it was mentioned earlier, k-connectivity is important for multi-path routing, reliability, and security in networks. By definition, a network is k-connected if there does not exist a set of $k-1$ vertices whose removal disconnects the graph. In particular, 1-connectivity ($k = 1$) is equivalent to connectivity. Using similar arguments, we find that the probability that $g(n, r)$ is not k-connected, $p_{k,disc}(n, r)$ is dominated by the probability that there exists at least one vertex in the network with degree less than k. In summary, the probability that $g(n, r)$ is not k-connected, can be approximated by

$$p_{k,disc}(n,r) \simeq \sum_{j=0}^{k-1} n\binom{n}{j} \int_{S_0} [\nu(B(\overline{X}, r(n)))]^j \times \Big(1 - \nu(B(\overline{X}, r(n)))\Big)^{n-j-1} dm(\overline{X}). \quad (262)$$

Figure 60, validates this expression for $k = 2$. Again we verify that the formula matches the

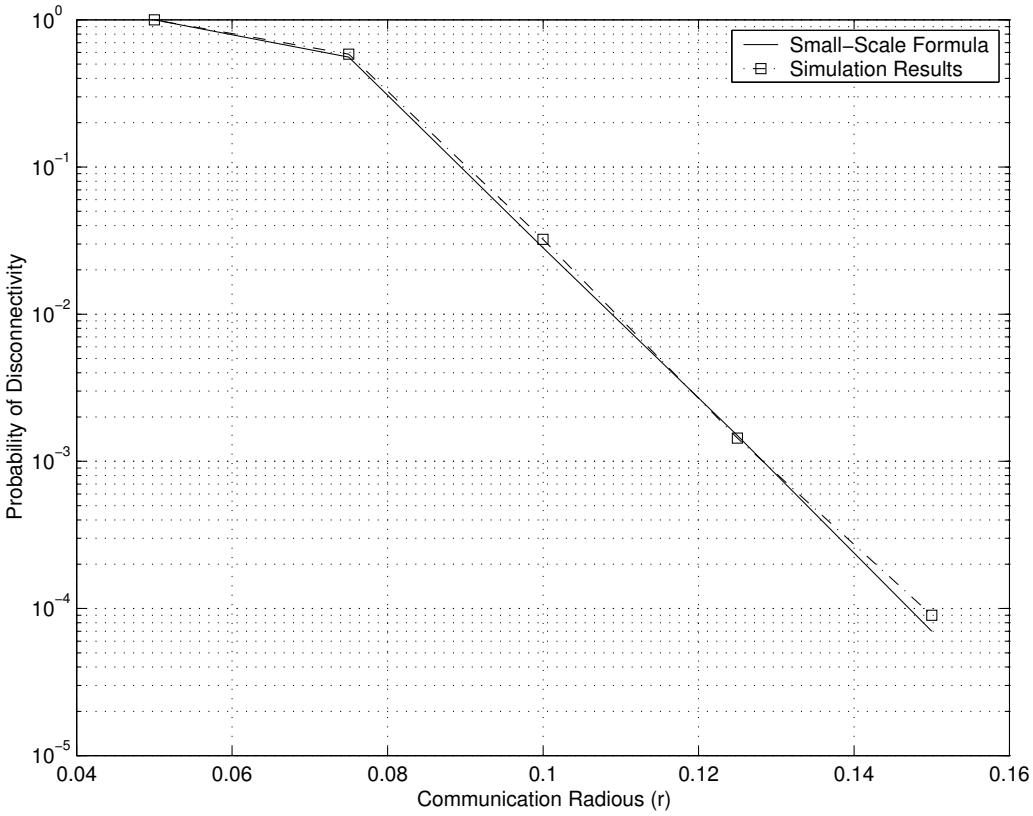

Figure 59: Disconnectivity probability of $g(500, r, 1)$ using (261) and simulation results.

simulation results. Our simulations for larger values of k consistently confirms the validity of (262). Here, we omit these results.

8.4.2 Effect of Constant Factors

So far, we have seen the importance of boundary effects in the analysis of finite networks. We now discuss another important issue. In asymptotic analysis, we usually neglect constant factors. However, in small-scale analysis, we must consider them. This is in fact, a difference between any finite analysis and asymptotic analysis and is not specific to geometric graphs. To show the importance of constant factors in the geometric graphs of wireless networks, we consider the medium access (MAC) layer capacity. Asymptotic MAC-layer capacity of ad hoc wireless networks is studied in [7]. The MAC-layer capacity is defined in [7] as the maximum possible number of concurrent transmissions at the media access layer. It is shown in [7] that for a wide class of MAC protocols including IEEE 802.11, the MAC-layer capacity can be modeled as a maximum Distance-2 matching (D2EMIS) problem in the

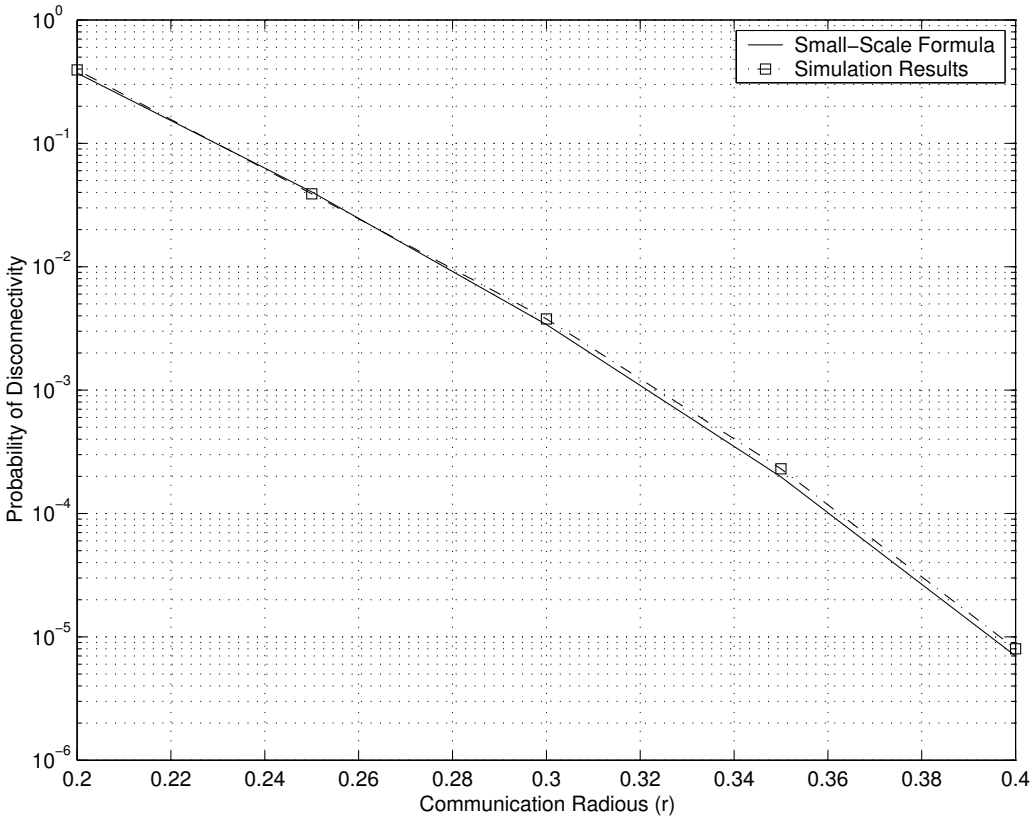

Figure 60: Probability that $g(100, r, 1)$ is not two-connected, using (261) and simulation results.

underlying wireless network. That is, given a graph $G(V, E)$, find a maximum set of edges $E' \subseteq E$ such that no two edges in E' are connected by another edge in E. It is shown in [7] that for $g(n, r)$, the MAC-layer capacity is optimized at $r = \Theta(\frac{1}{\sqrt{n}})$ and is given by $\Theta(n)$. Although this is an important and valuable result, it has very limited value when we consider finite networks. For example, suppose we have a network consisting of 100 sensors and we want to choose the communication radius such that the MAC-layer capacity is optimized. The asymptotic result does not tell us what the value of r should be. Moreover, we do not know what the optimum MAC-layer capacity would be. This example clearly shows the importance of constant factors in small-scale analysis. In the next section we analyze the average MAC-layer capacity for finite networks and obtain simple lower and upper bounds. Using these bounds, we try to answer the above question about the MAC-layer capacity of a finite sensor network.

Small-Scale Analysis of MAC-Layer Capacity of $g(n,r)$***:***

In this section we analyze the average MAC-layer capacity of $g(n,r)$, i.e the maximum number of possible concurrent transmissions which is available on average in $g(n,r)$. As it was mentioned, we find simple upper and lower bounds and by which we find the optimum value of r and the corresponding average MAC-layer capacity. We now prove the following lower bound. Let $MAC(n,r)$ be the average MAC-layer capacity of $g(n,r)$.

Theorem 45. *Define*

$$s = \int_{S_0} \nu(B(\overline{X}, 2r))dm(\overline{X}), \tag{263}$$

$$t = \frac{1-(1-s)^n}{s}. \tag{264}$$

Then, the average MAC-layer capacity satisfies

$$MAC(n,r) \geq t\Big(1-(1-\pi r^2)^{n-t}\Big). \tag{265}$$

Proof. The proof is constructive. That is, we use an algorithm to find a set of m concurrent transmissions in $g(n,r)$ such that on average m satisfies the lower bound given by the theorem. Here is the summary of the algorithm. We first find t *central* nodes $a_1, a_2, ..., a_t$ in the network located at $\overline{X}_{a_1}, \overline{X}_{a_2}, ..., \overline{X}_{a_t}$ with the following property. For any i, j the distance between a_i and a_j satisfies

$$\| \overline{X}_{a_i} - \overline{X}_{a_j} \| > 2r. \tag{266}$$

Thus, in particular the transmissions between a_i's and their neighbors results in a valid Distance-2 matching. Therefore, $MAC(n,r)$ is lower-bounded by the number of central nodes that have at least one neighbor. Then, we show a fraction $\big(1-(1-\pi r^2)^{n-t}\big)$ of central nodes have at least one neighbor. Thus, we conclude that $MAC(n,r) \geq t\big(1-(1-\pi r^2)^{n-t}\big)$.

Remember that in $g(n,r)$, n nodes are deployed independently and uniformly at random over S_0. Let Y be the number of central nodes obtained by our algorithm, and let Y_i be a random variable that takes value 1 if the i'th node in the network becomes a central node

and 0 otherwise. Then

$$Y = \sum_{i=1}^{n} Y_i. \tag{267}$$

We start with the first node in the network. The first node is always a central node. Hence, $EY_1 = \text{Prob}\{Y_1 = 1\} = 1$. The second node would be a central node if its distance from the node one is larger than $2r$. Thus, $EY_2 = \text{Prob}\{Y_2 = 1\} = 1 - s$, where s is the average value of $\nu(B(\overline{X_2}, 2r))$ which is given by (263). In general, we have

$$EY_k = \text{Prob}\{Y_k = 1\} \geq 1 - s(Y_1 + Y_2 + ... + Y_{k-1}). \tag{268}$$

By solving this recursive equation we obtain $EY \geq \frac{1-(1-s)^n}{s} = t$. Now, there are $n - t$ nodes left in the network which are not central. It is easy to see that, the probability that a central node does not have any neighbors is less than $\big(1 - \pi r^2)^{n-t}$. Thus, on average there are $t\big(1 - (1 - \pi r^2)^{n-t}\big)$ central nodes that have at least one neighbors. This concludes the theorem. □

We now obtain a simple upper bound on the average MAC-layer capacity.

Theorem 46. *Consider a wireless network graph $g(n, r)$. Define*

$$n_1 = n \int_{S_0} (1 - \nu(B(\overline{X}, r)))^{n-1} dm(\overline{X}). \tag{269}$$

Then, the average MAC-layer capacity satisfies

$$MAC(n, r) \leq \frac{n - n_1}{2 + 1.37r^2 n}. \tag{270}$$

Proof. Consider a maximum Distance-2 matching for $g(n, r)$ of size m. Consider three kinds of nodes. The first group, is the group of nodes involved in the matching. The number of these nodes is equal to $2m$. The second group is the set of nodes that are not in the matching but are neighbors of the nodes involved in the matching. The number of nodes in this group is shown by m_2. Finally, the third group is the set of isolated nodes. The average number of nodes in the third group is equal to n_1 given by (269). The upper bound is proved by noting that $2m + m_2 + n_1 \leq n$. It remains to find an estimate for m_2. To do this, for any node in the first group draw a circle centered at the node with radius $r/2$.

Note that this choice of radius ensures that the circles for any edge in the matching do not intersect with the circles for other edges in the matching. For any edge e in the matching let $s(e)$ be the area of the unions of the two corresponding circles. A simple integration shows the average value for $s(e)$ is $1.37r^2$. Therefore, any two nodes connected by an edge in the matching have on average $1.37r^2n$ neighbors. This implies that the average number of nodes in the second group is equal to $m_2 = m \times 1.37r^2n$. Thus we obtain $m \leq \frac{n-n_1}{2+1.37r^2n}$. □

Figure 61 shows the upper and lower bounds on the MAC-layer capacity of $g(100, r)$. The lower bound is maximized at $r = .06$, while the upper bound is maximized at $r = .08$. Thus, to optimize the MAC-layer capacity we can choose $.06 < r < .08$. We also note that the maximum achievable MAC-layer capacity is between 15 and 30. An interesting open problem is to tighten the bounds to obtain a more accurate estimate of MAC-layer capacity.

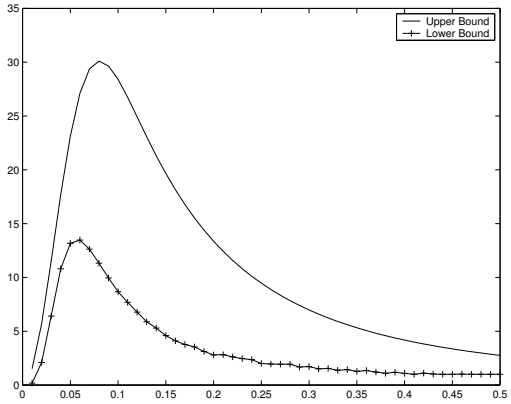

Figure 61: Upper and lower bounds on the average MAC-layer capacity of $g(100, r)$.

8.4.3 Lack of Concentration

In asymptotic analyses, usually random variables concentrate on their average values. Thus, it usually suffices to only determine the expected value. However, in small-scale analysis this is not the case. Thus, knowing the expected value is not usually enough. To clarify this, it is useful to consider geometric routing algorithms. A survey on routing protocols for wireless sensor networks can be found in [2]. Suppose nodes A and B are two fixed nodes on the plane that are located at the unit distance away from each other. In geometric routing when node A wants to send the data to node B, the information is usually sent hop

by hop to B. Each node passes the data to another node which is somewhat closer to the destination, node B. An example of such routing algorithms is GPSR [57]. Since in these algorithms the next hop is typically determined locally, when the density of nodes is large, by a martingale argument we can usually prove that the path length (the number of hops) is concentrated around its average with high probability. Thus, determining the average path length would be sufficient. On the other hand, for small-scale networks, the number of hops can considerably deviate from the average. Thus, analysis of geometric routing protocols for small-scale networks can be very important. To show this, we now analyze a simple geometric routing algorithm. We should note that the algorithm is not optimal in terms of energy, delay, etc. However, it is good enough to show the distinct characteristics of small-scale networks.

Consider the following scenario. As shown in Figure 62, let us assume that node A wants to send some information to node B. Suppose the nodes are distributed on the plane based on a Poisson distribution with density λ. Assume the distance between A and B is $d(A, B) = 1$. We connect A and B by a virtual line and also consider a virtual rectangle shown in Fig. 62 with width w. The routing path consists of all the nodes in the rectangle, from left to right. In this routing scenario, we assume that the packets travel from a node to its right neighbor node with the shortest horizontal distance. Assume that A is located at $(0, 0)$, and B at $(1, 0)$, and the ith node in the rout is located at (X_i, Y_i). Then, $X_{i+1} - X_i$ has an exponential distribution with parameter λw. Thus, if H is the number of hops from A to B, we have

$$\text{Prob}\{H = h\} = \text{Gamcdf}(1, h - 1, \frac{1}{w\lambda}) - \text{Gamcdf}(1, h, \frac{1}{w\lambda}),$$

where $\text{Gamcdf}(x, h, \eta)$ is the value of the Gamma distribution function with parameters h and η at point x. Figures 63, 64, and 65 show the distribution of H for different values of $\eta = \lambda w$. For small η, it is clear that the distribution is very wide. However for larger η, the distribution concentrates around its average, $EH = \eta + 1$. This shows that although in the asymptotic case the average value can suffice for the analysis, the whole distribution should be known in the finite density.

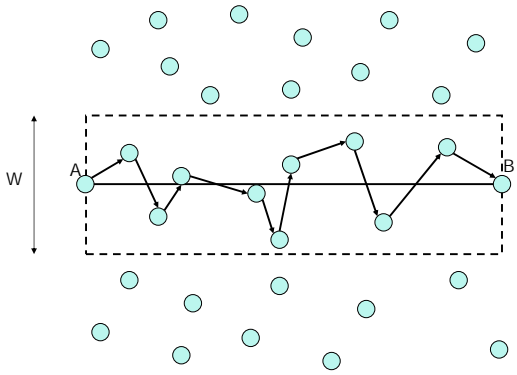

Figure 62: Illustration of a simple geometric routing.

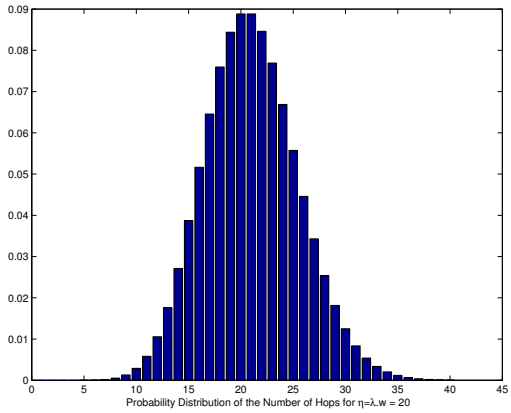

Figure 63: Probability distribution of the number of hops between nodes A and B, for $\eta = \lambda.w = 20$

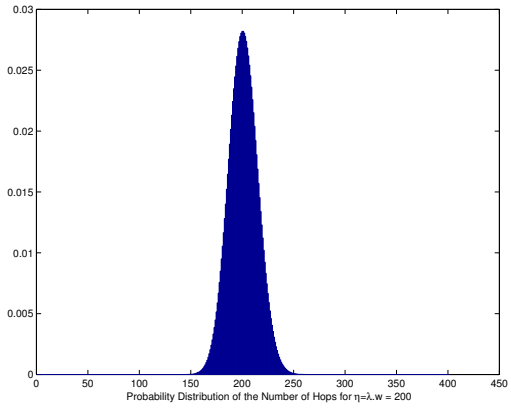

Figure 64: Probability distribution of the number of hops between nodes A and B, for $\eta = \lambda.w = 200$

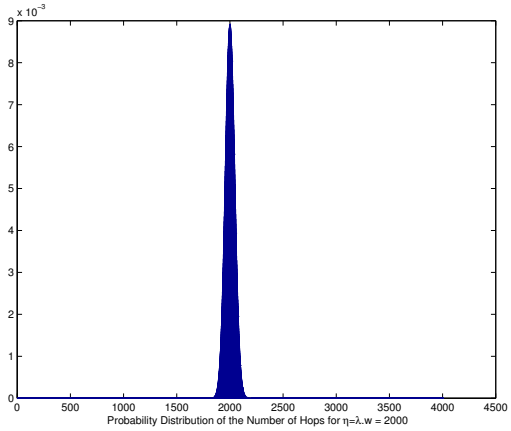

Figure 65: Probability distribution of the number of hops between nodes A and B, for $\eta = \lambda.w = 2000$

As an application of this, let us consider the energy issue. In wireless sensor networks, energy is arguably the most important constraint. Thus, we would like to minimize the energy consumption. We assume that the energy needed for a direct transmission from a node to a neighbor at distance d is proportional to d^2. Here, we assume that every sensor adjusts its transmission power according to its distance from the recipient node. Then, using the distribution of the random variables involved in our simple geometric routing, we conclude that the average total energy consumption in communication between A and B is proportional to

$$Avg.Energy \propto (\lambda w + 1)\left(\frac{2}{\lambda^2 w^2} + \frac{w^2}{12}\right). \tag{271}$$

Thus, for a given λ, we can find the value of w that minimizes the average energy. It is very important to note that the geometric algorithm used here is not the best possible, and the assumption of the energy adjustment in the sender node may not be realistic in some scenarios. However, almost all geometric routings have similar properties. Specifically, in all geometric algorithms, when the density of nodes tends to infinity, the number of hops converges to the average value, while when the network is not very dense the number of hops can deviate considerably.

8.5 Conclusion

In this chapter, we introduced small-scale analysis of wireless networks. We provided some compelling evidence to show that asymptotic results are not suitable for analyzing practical finite networks. We established a framework for small-scale analysis. We considered connectivity, coverage, MAC-layer capacity and routing algorithms of finite networks. We obtained a very simple formula for connectivity of wireless networks and verified it by simulation results. The formula was then extended to include k-connectivity. We then studied MAC-layer capacity and obtained simple lower and upper bounds. Using these bounds we estimated the optimum value for achieving the highest capacity. Finally, we studied geometric routings. Using, these examples we confirmed that finite-scale networks posses unique characteristics that require a new framework distinct from asymptotic approaches.

This chapter opens up many research possibilities that offer potential for further research. In the past, many other important properties of wireless networks have been studied for large-scale networks. It is an important task to extend these results for networks with practical sizes, i.e., small-scale networks. For example, there are several other measures for network capacity such as transport capacity, information theoretic capacity, and capacity of cooperative nodes. Asymptotic analysis of these definitions has been studied extensively. It is very useful to extend these results to small-scale networks.

Small-scale analysis can reveal the effects of network parameters on networks characteristics. A next step would be to use the small-scale framework in the design, analysis, and evaluation of communication algorithms for wireless networks.

CHAPTER IX

CONCLUSION

This work introduced and explored new theoretical and practical issues in the study of low-density parity-check (LDPC) codes and wireless networks. Both LDPC codes and wireless networks were studied in the context of random graphs.

For LDPC codes, this work studied a variety of theoretical and practical problems. First, it introduced an improved decoding algorithm for LDPC codes that would be specifically suitable for practical applications in which we cannot use large block lengths. It was shown that the algorithm significantly outperformed the standard iterative decoding algorithm. Second, it studied rate-compatible LDPC codes. It provided their fundamental properties, discussed their design, and showed how they might be used to solve an important open problem relating capacity-achieving LDPC codes. Third, this work introduced a new class of LDPC codes for non-uniform error correction and showed that the new coding scheme could significantly increase the storage capacity of volume holographic memory systems. Finally, it provided a variety of performance bounds for LDPC codes.

In wireless networks, this work focused on sensor and ad hoc networks. It introduced small-scale analysis of wireless networks. It provided compelling evidence to show that asymptotic results are not suitable for analyzing practical finite networks. In particular, it considered connectivity, coverage, MAC-layer capacity and routing algorithms of finite networks. Using, these examples it confirmed that finite-scale networks posses unique characteristics that require a new framework distinct from asymptotic approaches. Finally, this work studied connectivity properties of large-scale networks. Specifically, it was shown how the unreliability of nodes and links, and the non-uniform distribution of the nodes could affect the connectivity properties of sensor networks.

This work opens up many research possibilities that offer potential for further research. For example, there are many issues that need to be investigated in analysis of finite networks.

In the past, many other important properties of wireless networks have been studied for large-scale networks. It is an important task to extend these results for networks with practical sizes, i.e., small-scale networks. For example, there are several other measures for network capacity such as transport capacity, information theoretic capacity, and capacity of cooperative nodes. Asymptotic analysis of these definitions has been studied extensively. It is very important and useful to extend these results to small-scale networks.

There are many possible directions for further research on LDPC codes, too. Here, we briefly describe some of the several interesting and potentially rich open directions for further research.

This work studied LDPC codes for data frames and developed a general coding scheme suitable for frames. This scheme allows unequal error protection, separate decoding of headers without the need to decode the rest of the frame, and some other desirable properties. It would be very interesting to consider the application of this coding scheme to practical scenarios such as optical networks and data file transfer applications. It is also interesting to analyze the tradeoff between computational complexity and average performance of these codes for the purpose of error detection.

This work also introduced improved decoding algorithms for LDPC codes. Many of the results were based on simulations, and it is an interesting open problem to find analytical results.

Coding for data storage is another interesting topic. In particular, this work presented new coding schemes for holographic data storage systems that resulted in more than fifty percent increase in the storage capacity compared to previous methods. We believe the same ideas may be applicable to other data storage systems.

On the theoretical side, the most important theoretical open problem in LDPC coding is to prove that these codes actually achieve the capacity of general (other than the erasure channel) symmetric channels. This work already made progress by proving that if LDPC codes achieve the channel capacity when the code rate tends to zero, then they achieve the channel capacity for all rates. Thus, this problem and related issues would be an important direction for the future research.

APPENDIX A

SUPPLEMENTARY FOR CHAPTER 3

Average Number of Cycles in Tanner Graphs of LDPC Codes

In this appendix, we calculate the average number of cycles in a Tanner graph of LDPC codes. For clarity of exposition we perform the computations for regular graphs. The generalization to irregular graphs is trivial. Here we use the same ensemble described in [112]. To each variable or check node we assign d_v or d_c sockets respectively. We label the variable nodes and check nodes separately with the set $\{1, 2, ..., nd_v\}$. We then pick a random permutation π on $E = nd_v$ letters. For each i, we put an edge between the socket i and $\pi(i)$. Let $v_1, v_2, ..., v_l$ be l arbitrary variable nodes and $c_1, c_2, ..., c_l$ be l arbitrary check nodes. Let $s_{v_1}, s_{v_2}, ..., s_{v_l}$ be l sockets such that s_{v_j} belongs to v_j and let $s_{c_1}, s_{c_2}, ..., s_{c_l}$ be l sockets such that s_{c_j} belongs to c_j. Then the probability that these sockets construct the cycle $v_1 - c_1 - v_2 - c_2 ... - c_l - v_1$ is equivalent to:

$$\frac{1}{E} \times \frac{1}{E-1} ... \times \frac{1}{E-2l+1} \tag{272}$$

Since any variable node has d_v sockets and any check node has d_c sockets, the probability of having the cycle $v_1 - c_1 - v_2 - c_2 ... - c_l - v_1$ in the Tanner graph is

$$\left\{ \frac{[d_v d_c (d_v - 1)(d_c - 1)]^l}{E \times (E-1) ... (E-2l+1)} \right\}. \tag{273}$$

Therefore, the probability that there exists a cycle of length l with vertices $v_1, v_2, ...v_l$ and $c_1, c_2 ... c_l$ is

$$\frac{l!(l-1)!}{2} \left\{ \frac{[d_v d_c (d_v - 1)(d_c - 1)]^l}{E \times (E-1) ... (E-2l+1)} \right\}. \tag{274}$$

This proves (40). Evaluating (40) for a constant value of l results in (68). Using the following equations:

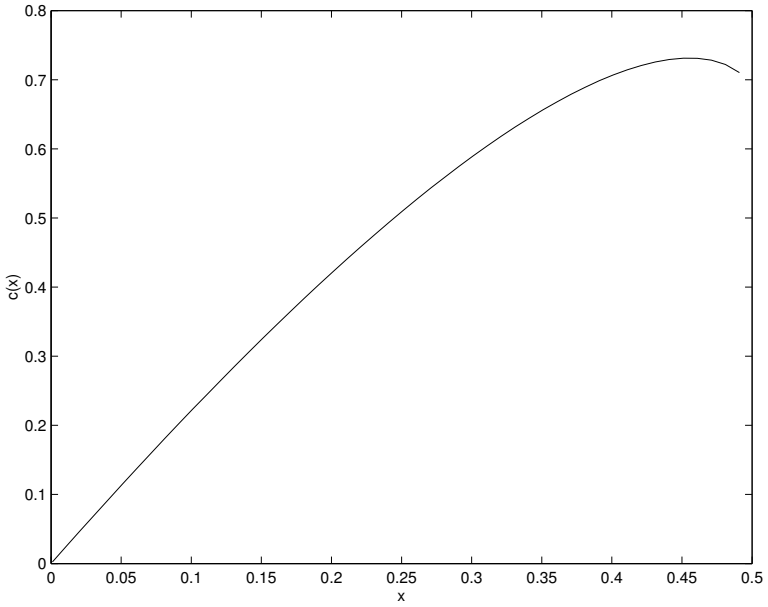

Figure 66: Cycle distribution in $g(d_v, d_c)$

$$\lim_{n\to\infty} \frac{1}{n} \ln \binom{n}{n\theta} = H(\theta),$$
$$\lim_{n\to\infty} \left[\ln(n) - \frac{1}{n} \ln(n!) \right] = 1,$$
$$\lim_{n\to\infty} \left[\frac{1}{n} \ln[n(n-1)...(n-n\theta+1)] - \theta \ln(n) \right] = H(\theta) + \theta \ln(\frac{\theta}{e}),$$

we get (42). Figure 66 shows the function $c(\theta)$ for $g(3, 6)$. Note that in the above model double edges are allowed. Therefore, we may have double cycles. However one can show that the probability of having a double cycle is very small.

Proof of Lemma 8 of Chapter A

The lemma can be proved in several ways. However, we found the following proof more insightful. The function $\rho(x)$ has the following properties.

$$\rho(0) = 0, \quad \rho(1) = 1, \quad \rho^{(n)}(x) > 0 \ \text{ for } \ x \in [0, 1] \text{ and } \ 0 \le n \le d_{c_{max}} \tag{275}$$

where $\rho^{(n)}(x)$ is the n'th derivative of the function ρ. Thus, the curve for the ρ looks like what is shown in Figure 67. The tangent line to the curve at point $(1, 1)$ is also shown in the figure. We have

$$1 - x_0 = \frac{1}{\rho'(1)}. \tag{276}$$

Therefore, the area of the triangle ABC is equal to $\frac{1}{2\rho'(1)}$. Since $\rho(x)$ is a convex function in $[0, 1]$, the area of the the triangle ABC is less than the area under the curve. That is, we have

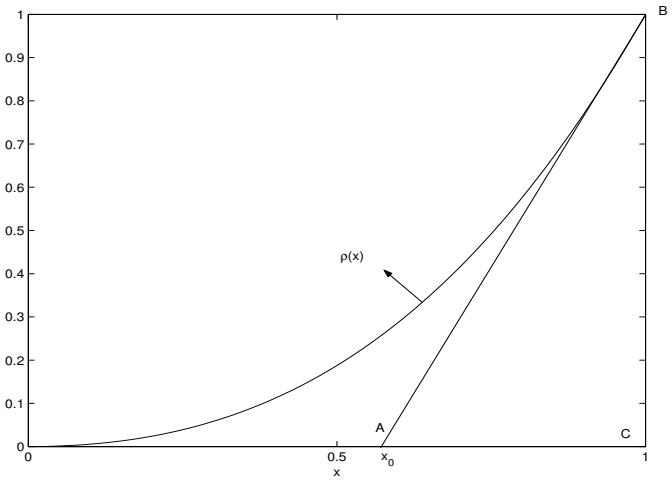

Figure 67: The function $\rho(x)$

$$\frac{1}{2\rho'(1)} < \int_0^1 \rho(x)dx. \tag{277}$$

Since $\lambda_2\rho'(1) < 1$ we obtain

$$\lambda_2 < 2\int_0^1 \rho(x)dx. \tag{278}$$

Thus, we conclude

$$\lambda_2' = \frac{\frac{\lambda_2}{2}}{\int_0^1 \lambda(x)dx} < \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - R. \tag{279}$$

APPENDIX B

SUPPLEMENTARY FOR CHAPTER 5

Gaussian Approximation for Analysis of the Ensemble $g(\Lambda, \rho)$

If the subchannels in Fig. 14 are BIAWGN, it is possible to use a Gaussian approximation similar to [25]. This method is useful for designing codes for VHM systems and for finding optimal puncturing distributions over the Gaussian channels. Here, we give the Gaussian approximation formulas for the $g(\Lambda, \rho)$. We use the function ϕ which is defined in [25]. Let $m_u^{(l)}$ denote the mean of messages from the check nodes to variable nodes in the l'th iteration. Let also $m_0^{(j)} = \frac{2}{\sigma_j^2}$ where σ_j is the variance of the noise in channel C_j in Fig. 14. Then we have

$$m_u^{(l)} = \sum_d \rho_d \phi^{-1}\left(1 - \left[1 - \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(m_0^{(j)} + (i-1) m_u^{(l-1)})\right]^{(d-1)}\right) \tag{280}$$

where $q^{(j)} = \frac{|E^{(j)}|}{|E|}$. Similar to [25] we define

$$f_d(\underline{s}, t) = \phi^{-1}\left(1 - \left[1 - \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(s^{(j)} + (i-1)t)\right]^{(d-1)}\right) \tag{281}$$

$$f(\underline{s}, t) = \sum_d \rho_d f_d(\underline{s}, t) \tag{282}$$

for $0 \leq t < \infty$. We can rewrite (280) as

$$t_l = f(\underline{s}, t_{l-1}) \tag{283}$$

where $\underline{s} = (s^{(1)}, s^{(2)}, ..., s^{(k_r)}) = (m_0^{(1)}, m_0^{(2)}, ..., m_0^{(k_r)})$ and $t_l = m_u^{(l)}$, and $t_0 = 0$. Similar to [25] one can show that $t_l(s)$ converges to infinity if and only if $t < f(\underline{s}, t)$ for all $t \in \mathbb{R}^+$. An equivalent formulation can be made by the following change of the variable

$$r_l = \sum_{j,i} q^{(j)} \lambda_i^{(j)} \phi(s^{(j)} + (i-1) t_l). \tag{284}$$

We also define

$$h_i^{(j)}(s,r) = \phi\big(s + (i-1)\sum_d \rho_d \phi^{-1}([1-(1-r)^{(d-1)}])\big) \tag{285}$$

$$h(\underline{s},r) = \sum_{j,i} \lambda_i^{(j)} q^{(j)} h_i^{(j)}(s^{(j)},r) \tag{286}$$

Then, we have

$$r_l = h(\underline{s}, r_{l-1}) \tag{287}$$

where $\underline{s}= (s^{(1)}, s^{(2)}, ..., s^{(k_r)}) = (m_0^{(1)}, m_0^{(2)}, ..., m_0^{(k_r)})$ and $r_0 = \sum_j q^{(j)}\phi(s^{(j)})$. Again $r_l(s) \longrightarrow 0$ if and only if $r > h(s,r)$ for all $r \in (0, r_0)$. It is easy to show that $r_l(s) \longrightarrow 0$ if and only if $r > h(s,r)$ for all $r \in (0,1)$. This fact is useful when we use linear programming for the optimization of the degree distribution or puncturing pattern.

APPENDIX C

SUPPLEMENTARY FOR CHAPTER 6

Proof of Theorem 21 of Chapter 6

For simplicity we prove the theorem for random puncturing. The random puncturing scheme can be modelled as transmitting over a channel C_{eq} as shown in Fig 37 with the following description. Assume a bit is transmitted through C_{eq}. Then, with probability p, this bit is transmitted over a channel with zero capacity and with probability $1-p$ it is transmitted over the channel C_θ. Now, using similar discussion to the proof of Theorem 6, we can show that if in the above model we replace C_θ with the channel $C_{\theta_{min}}$, which is a channel of capacity one, the overall error probability decreases. On the other hand, replacing C_θ with the channel $C_{\theta_{min}}$ in Figure 37, the channel C_{eq} becomes equivalent to a BEC with erasure probability $\epsilon = p$. This is because every bit is either transmitted through the channel with zero capacity C_2 (with probability p) or through the noiseless channel $C_{\theta_{min}}$. This proves a lower bound on the error probability that results in the upper bound on the puncturing threshold.

Examining the above discussion indicates that the theorem applies when the error rate is averaged over all bits, i.e., both punctured and unpunctured bits. A more realistic case is to consider the error rate of only unpunctured bits. By a similar argument to Lemma 1 in [106], for $\theta > \theta_{min}$, if the error rate of punctured bits is bounded away from zero, then the error rate of unpunctured bits is bounded away from zero as well. This implies the following corollary.

Corollary 15. *Consider the ensemble of LDPC codes defined by the pair (λ, ρ). Let (λ, ρ, p) be the ensemble of LDPC codes that are generated by randomly puncturing of the ensemble (λ, ρ) by the puncturing fraction p. Assume a randomly chosen code from the ensemble (λ, ρ, p) is used over the channel C_θ, with $\theta > \theta_{min}$. Let ϵ_{th} be the threshold of the ensemble*

(λ, ρ) for BEC under the iterative decoding. If $p > \epsilon_{th}$, then the error probability of decoding the punctured code is bounded away from zero independent of the communication channel.

The above discussion gives upper bounds on the puncturing fraction of LDPC codes. We will now show that these upper bounds are actually the puncturing threshold of LDPC codes in the sense that if the puncturing fraction is less than the upper bounds then the punctured LDPC code is an asymptotically good code. We first prove a lemma.

Lemma 24. *Consider the ensemble of LDPC codes defined by the pair (λ, ρ) that are randomly punctured by the puncturing fraction $p < p_{th}$ where $p_{th} = \epsilon_{th}$ is the upper bound given by Corollary 15. Then there exists a $\theta_1 > \theta_{min}$ such that if the channel parameter θ is smaller than θ_1, the punctured ensemble satisfies the stability condition.*

Proof. By the assumption of the lemma, a randomly chosen code from the ensemble (λ, ρ) can be used to obtain arbitrarily small bit error rate over a BEC with erasure probability $\epsilon = p$. Thus, using the stability condition [54], we have

$$\lambda_2 \rho'(1) < \frac{1}{p}. \tag{288}$$

Now consider the ensemble of LDPC codes defined by (λ, ρ) that are randomly punctured by a puncturing fraction p. Suppose a code from the punctured ensemble is used over a channel with the parameter θ. Then, assuming the all-one codeword has been sent, the density of the LLR's from the channel is equal to

$$f_0(x) = p\delta(x) + (1-p) f_{z_\theta}(x) \tag{289}$$

We need to show that for suitably chosen $\theta > \theta_{min}$ we have

$$\lambda_2 \rho'(1) < e^r, \quad r = -\ln\left(\int_{\mathbb{R}} f_0(x) e^{-\frac{x}{2}} dx \right) \tag{290}$$

Since as θ goes to θ_{min}, $f_{z_\theta}(x)$ converges to $\Delta_\infty(x)$ (in the sense defined in [54]). By choosing $\theta > \theta_{min}$ small we can make the integral $\int_{\mathbb{R}} (1-p) f_{z_\theta}(x) e^{-\frac{x}{2}} dx$ arbitrarily small. Thus, using (288) and (289) we conclude that there exists a $\theta > \theta_{min}$ for which we have $\lambda_2 \rho'(1) < e^r$ for $r = -\ln\left(\int_{\mathbb{R}} f_0(x) e^{-\frac{x}{2}} dx \right)$. □

Now we show under the conditions of Lemma 24, there exists a $\theta^* > \theta_{min}$ such that if the channel parameter θ is smaller than θ^*, then

$$\lim_{l\to\infty} P_e(F_l) = 0. \tag{291}$$

For the given ensemble, the probability density function f_l can be written as

$$f_l(x) = y_l\delta(x) + (1 - y_l)g_l(x) \tag{292}$$

where y_l satisfies

$$\begin{aligned} y_0 &= p \\ y_l &= p\lambda\big(1 - \rho(1 - y_{l-1})\big). \end{aligned} \tag{293}$$

By the conditions of the theorem

$$\lim_{l\to\infty} y_l = 0. \tag{294}$$

Moreover, for a fix value of l, we have

$$\lim_{\theta\to\theta_{min}} P_e(g_l) = 0. \tag{295}$$

Now, by Lemma 24, the stability condition is satisfied, for $\theta < \theta_1$. Thus, by the stability theorem in [54], there exists a constant $\zeta > 0$ such that if $P_e(F_l) < \zeta$, for some $l \in \mathbb{N}$, then $P_e(F_l)$ converges to zero as l tends to infinity. Choose l_1 large enough such that $y_{l_1} < \zeta$. Now fix l_1 and choose $\theta_2 > \theta_{min}$ such that $P_e(g_{l_1}) < \zeta/2$. Thus for $\theta < min(\theta_1, \theta_2)$, the stability condition is satisfied and we have

$$P_e(F_{l_1}) = \frac{1}{2}y_{l_1} + (1 - y_{l_1})P_e(g_{l_1}) < \zeta. \tag{296}$$

Therefore, $P_e(F_l)$ converges to zero as l tends to infinity.

APPENDIX D

SUPPLEMENTARY FOR CHAPTER 7

Proof of Theorem 31 of Chapter 7

Proof. Define $\omega(n) := n\pi r^2(n)p_e(n) - \ln(n)$, thus $\pi r^2(n) = \frac{\ln n + \omega(n)}{np_e(n)}$. Let $S_1 = \overline{S(\overline{O}, 1-2r(n))}$. We now obtain

$$\begin{aligned} EZ_n = & \\ & n \int\limits_{S_0} \left(1 - \nu(B(\overline{X}, r(n)))p_e(n)\right)^{n-1} dm(\overline{X}) \\ \geq & n \int\limits_{S_1} \left(1 - \nu(B(\overline{X}, r(n)))p_e(n)\right)^{n-1} dm(\overline{X}) \\ = & n \int\limits_{S_1} \left(1 - \frac{\ln n + \omega(n)}{n}\right)^{n-1} dm(\overline{X}) \\ = & n \left(1 - \frac{\ln n + \omega(n)}{n}\right)^{n-1} m(S_1) \\ = & e^{-\omega(n)}(1 + o(1)). \end{aligned} \tag{297}$$

Therefore, we conclude that $\lim\limits_{n\to\infty} EZ_n(r(n)) = \infty$ if $\lim\limits_{n\to\infty} \omega(n) = -\infty$. Moreover, $\lim\limits_{n\to\infty} EZ_n(r(n)) > 0$ if $\lim\limits_{n\to\infty} \omega(n) < \infty$. Now assume that $\lim\limits_{n\to\infty} \omega(n) > -\infty$. Let $Y_{3,n}$ be the number of isolated vertices in S_3. Then we get

$$\begin{aligned} EY_{3,n} & \leq n r^2(n) \left(1 - \frac{\pi r^2(n)}{4} p_e(n)\right)^{n-1} \\ & \leq n r^2(n) e^{-\frac{\pi r^2(n)}{4} p_e(n)(n-1)}. \end{aligned} \tag{298}$$

Using $p_e(n) \geq \frac{c}{\ln n}$ and $\pi r^2(n) = \frac{\ln n + \omega(n)}{np_e(n)}$, we conclude

$$EY_{3,n} = O(\frac{\ln n(\ln n + \omega(n))e^{-\omega(n)/4}}{n^{\frac{1}{4}}}) = o(1). \tag{299}$$

Therefore, there is no isolated vertex in S_3 with high probability. Next, let $Y_{2,n}$ be the

number of isolated vertices in S_2. Then

$$EY_{2,n} = n \int_{S_2} \left(1 - \nu(B(\overline{X}, r(n)))p_e(n)\right)^{n-1} dm(\overline{X}) \tag{300}$$

Using the Laplace method for integrals and Lemma 21, it can be shown that

$$EY_{2,n} = O(\frac{e^{-\frac{\omega(n)}{2}}}{r(n)p_e(n)\sqrt{n}}) \tag{301}$$

Using $p_e(n) \geq \frac{c}{\ln n}$ and $\pi r^2(n) = \frac{\ln n + \omega(n)}{np_e(n)}$, we conclude

$$EY_{2,n} = O(\frac{e^{-\frac{\omega(n)}{2}}}{\sqrt{(c + \frac{c\omega(n)}{\ln(n)})}}). \tag{302}$$

Thus if $\lim_{n\to\infty} \omega(n) = \infty$ then $Y_{2,n} = 0$ asymptotically almost surely. Moreover, if $0 < \lim_{n\to\infty} \omega(n) < \infty$ then $Y_{2,n}$ is finite asymptotically almost surely. Combining with (297) we conclude the theorem. □

Proof of Theorem 32

Proof. By Theorem 31, when $\lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] = \infty$, we have $\lim_{n\to\infty} EZ_n(r(n)) = 0$. Thus, by Markov's inequality there is no isolated vertex with high probability. Then, by Theorem 30 the graph is connected asymptotically almost surely. Hence, we focus on the proof of the other direction. That is if $0 < \lim_{n\to\infty} \left[n\pi r^2(n)p_e(n) - \ln(n)\right] < \infty$ (or equivalently $0 < \lim_{n\to\infty} EZ_n(r(n)) < \infty$) , then there exists $\delta > 0$ such that $\liminf_{n\to\infty} p_n^{disc} > \delta > 0$, where p_n^{disc} is the probability that g_n is disconnected. The proof is as follows. Let $A_{n,j}$ be the event that the vertex v_j is isolated. Then we want to prove

$$\limsup_{n\to\infty} \Pr\{\bigcap_{i=1}^{n} \overline{A_{n,i}}\} < 1. \tag{303}$$

To prove the above, we use Lemma 22. Let $\Delta_n = \sum_{i=1}^{n} \sum_{j\neq i} \Pr\{A_{n,i} \cap A_{n,j}\}$. We show that under the condition $0 < \mu < \infty$, we have $\lim_{n\to\infty} \Delta_n = \Delta < \infty$. Thus by applying Lemma 22 we conclude the theorem. It remains to prove $\Delta < \infty$. We note that

$$\begin{aligned} \Delta_n \leq n(n-1) \int_{S_0 \times S_0} \Big(&1 - \nu(B(\overline{X}, r(n)))p_e(n) - \\ &\nu(B(\overline{X}, r(n)))p_e(n) + \\ &\nu(B(\overline{X}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)\Big)^{n-2} dm(\overline{X}) \times m(\overline{Y}) \end{aligned} \tag{304}$$

We have $S_0 \times S_0 = (S_1 \times S_1) \cup (S_0 \times S_0 \setminus S_1 \times S_1)$. It suffices to show that the integral over the set $S_1 \times S_1$ and $S_0 \times S_0 \setminus S_1 \times S_1$ is finite. Let Δ_n^1 and Δ_n^2 be the two integrals respectively. For example, for $S_1 \times S_1$ we have

$$\begin{aligned}
\Delta_n^1 &= n(n-1) \int\limits_{S_1 \times S_1} \Big(1 - \nu(B(\overline{X}, r(n)))p_e(n) - \\
&\quad \nu(B(\overline{X}, r(n)))p_e(n) + \\
&\quad \nu(B(\overline{X}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)\Big)^{n-1} d(m \times m) \\
&= n(n-1) \int\limits_{S_1 \times S_1} \Big(1 - \frac{\ln n + \omega(n)}{n} - \\
&\quad \frac{\ln n + \omega(n)}{n} + \\
&\quad \nu(B(\overline{X}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)\Big)^{n-1} d(m \times m) \\
&\leq e^{-2\omega(n)} \int\limits_{S_1 \times S_1} e^{\nu(B(\overline{X}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)(n-1)} d(m \times m) \\
&= e^{-2\omega(n)} \int\limits_{S_1} e^{\nu(B(\overline{O}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)(n-1)} dm(Y) \\
&= e^{-2\omega(n)} \int\limits_{S_1 \cap B(\overline{O}, 2r(n))} e^{\nu(B(\overline{O}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)(n-1)} dm(Y) \\
&\quad + e^{-2\omega(n)} \int\limits_{B(\overline{O}, 2r(n))} e^{\nu(B(\overline{O}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)(n-1)} dm(Y) \\
&= e^{-2\omega(n)} + e^{-2\omega(n)} \int\limits_{B(\overline{O}, 2r(n))} e^{\nu(B(\overline{O}, r(n))) \cap B(\overline{Y}, r(n)))p_e^2(n)(n-1)} dm(Y).
\end{aligned} \tag{305}$$

Using the Laplace method for integrals and Lemma 21 we obtain

$$\Delta_n^1 = e^{-2\omega(n)} + O\left(\frac{e^{-(2-p_e(n))\omega(n)}}{n^{(2-p_e(n))} p_e(n)^4 r^2(n)}\right) \tag{306}$$

Using $p_e(n) \geq \frac{c}{\ln n}$ and $0 < \lim\limits_{n\to\infty} \omega(n) < \infty$, we conclude

$$\lim_{n\to\infty} \Delta_n^1 < \infty. \tag{307}$$

Similarly, we can show $\lim\limits_{n\to\infty} \Delta_n^2 < \infty$. Therefore, $\lim\limits_{n\to\infty} \Delta_n = \Delta < \infty$, which concludes the theorem. □

Proof of Theorem 35

Proof. By a simple coupling argument, we find that the probability of having at least one isolated vertex is a decreasing function of $r(n)$. If $\alpha < 1$, then for any constant c and large

enough n, we have

$$r(n) < \sqrt{\frac{\ln n + c}{\pi n p_e(n)}}. \tag{308}$$

Thus, by Theorem 33, the probability that $g = g(n, r, p_e)$ has at least one isolated vertex is asymptotically greater than or equal to $e^{-e^{-c}}$ for any real number c. Thus, if $\alpha < 1$, the graph $g = g(n, r, p_e)$ has an isolated vertex with high probability, and thus it is not k-connected for any positive integer k.

Now, by Theorem 30, it suffices to prove that if $\alpha > 1$, for any fixed $k \in \{0, 1, 2, ...\}$, $g(n, r, p_e)$ does not have any vertices of degree k with high probability. Let $\alpha > 1$ and $Y_{j,k,n}$ be the number of vertices of degree k in S_j, for $j = 1, 2, 3$. It suffices to show $Y_{j,k,n} = 0$ asymptotically almost surely for $j = 1, 2, 3$.

We first consider $Y_{1,k,n}$. We have

$$\begin{aligned} &EY_{1,k,n} = \\ &n \int_{S_1} \binom{n}{k} [\nu(B(\overline{X}, r(n)))p_e(n)]^k \\ &\left(1 - \nu(B(\overline{X}, r(n)))p_e(n)\right)^{n-k-1} dm(\overline{X}). \end{aligned} \tag{309}$$

But for $\overline{X} \in S_1$, we have $\nu(B(\overline{X}, r(n))) = \pi r^2(n)$. Thus

$$EY_{1,k,n} = O\big(\frac{(\ln n)^k}{n^{\alpha-1}}\big) = o(1). \tag{310}$$

Therefore, $Y_{1,k,n} = 0$ asymptotically almost surely. We now consider $Y_{2,k,n}$. We have

$$\begin{aligned} &EY_{2,k,n} = \\ &n \int_{S_2} \binom{n}{k} [\nu(B(\overline{X}, r(n)))p_e(n)]^k \\ &\left(1 - \nu(B(\overline{X}, r(n)))p_e(n)\right)^{n-k-1} dm(\overline{X}). \end{aligned} \tag{311}$$

Using the Laplace method for integrals, Lemma 21, and $p_e(n) \geq \frac{c}{\ln n}$ we can write

$$EY_{2,k,n} = O\big(\frac{(\ln n)^{2k+1}}{n^{\frac{\alpha}{2} - \frac{1}{2} - o(1)}}\big) = o(1) \tag{312}$$

This implies $Y_{2,k,n} = 0$ asymptotically almost surely. We now prove $Y_{3,k,n} = 0$ asymptotically almost surely. We note that

$$\begin{aligned} & EY_{3,k,n} \leq \\ & nr^2(n)\binom{n}{k}(\pi r^2(n)p_e(n))^k\left(1-\frac{\pi r^2(n)}{4}p_e(n)\right)^{n-k-1} \\ & \leq n^{k+1}r^{2k+2}(n)p_e(n)e^{-\frac{\pi r^2(n)}{4}p_e(n)(n-k-1)}. \end{aligned} \tag{313}$$

Using $p_e(n) \geq \frac{c}{\ln n}$ and $\lim_{n\to\infty}\left(\frac{n\pi r^2(n)p_e(n)}{\ln n}\right) = \alpha$, we conclude

$$EY_{3,k,n} = O(\frac{(\ln n)^{k+2}}{n^{\frac{1}{4}-o(1)}}) = o(1) \tag{314}$$

This implies that $Y_{3,k,n} = 0$ asymptotically almost surely. □

REFERENCES

[1] Available online at http://lthcwww.epfl.ch/research/ldpcopt/.

[2] Akkaya, K. and Younis, M., "A survey on routing protocols for wireless sensor networks," *Elsevier Ad Hoc Network Journal*. To appear.

[3] Akkaya, K. and Younis, M., "A survey of routing protocols in wireless sensor networks," *Elsevier Ad Hoc Network Journal*, 2004. to appear.

[4] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, August 2002.

[5] An, X., Burr, G. W., and Psaltis, D., "Thermal fixing of 10,000 holograms in $linbo_3 : Fe$," *Appl. Opt.*, vol. 38, pp. 386–393, 1999.

[6] Ayanoglu, E., Gitlin, C., and Mazo, J., "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Transactions on Communications, 41(11):1677–1686, November 1993.*, vol. 41, pp. 1677–1686, 1993.

[7] Balakrishan, H., Barrett, C. L., Kumar, V. S. A., Marathe, M. V., and Thite, S., "The distance-2 matching problem and its relationship to MAC-layer capacity of ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, vol. 22, pp. 1069 –1079, 2004.

[8] Barak, O., Burshtein, D., and Feder, M., "Bounds on achievable rates of LDPC codes used over the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2483–2492, 2004.

[9] Berlekamp, E. R., McEliece, R. J., and van Tilborg, H. C. A., "On the inherent intractibility of certain coding problems," *IEEE Trans. Inform. Theory*, vol. 24, pp. 384–386, 1978.

[10] Bollobás, B., *Random Graphs*. Cambridge University Press, second ed., 2001.

[11] Booth, L., Bruck, J., Franceschetti, M., and Meester, R., "Covering algorithms, continuum percolation and the geometry of wireless networks," *Annals of Applied Probability*, vol. 13, May 2003.

[12] Booth, L., Bruck, J., M.Cook, Franceschetti, M., and Meester, R., "Continuum percolation with unreliable and spread out connections.," submitted.

[13] Boyarinov, I. and Katsman, G., "Linear unequal error protection code," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 168–175, 1981.

[14] Brady, D. J. and Psaltis, D., "Control of volume holograms," *J. Opt. Soc. Am. A*, vol. 9, pp. 1167–1182, 1992.

[15] Braginsky, D. and Estrin, D., "Rumor routing algorithm for sensor networks," *ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[16] Burr, G. W., Ashley, J., Coufal, H., Grygier, R. K., Hoffnagle, J. A., Jefferson, C. M., and Marcus, B., "Modulation coding for pixel-matched holographic data storage," *Opt. Lett.*, vol. 22, pp. 639–641, 1997.

[17] Burr, G. W., Jefferson, C. M., Coufal, H., Jurich, M., Hoffnagle, J. A., Macfarlane, R. M., and Shelby, R. M., "Volume holographic data storage at areal density of 250 gigapixels/in2," *Opt. Lett.*, vol. 26, pp. 444–446, 2001.

[18] Burshtein, D., Krivelevich, M., Litsyn, S., and Miller, G., "Upper bounds on the rate of LDPC codes," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2437–2449, 2002.

[19] Burshtein, D. and Miller, G., "Asymptotic enumeration method for analyzing LDPC codes," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1115–1131, 2004.

[20] Chan, H., Perrig, A., and Song, D., "random key predistribution schemes for sensor networks," in *2003 IEEE Symposium on Research in Security and Privacy*, pp. 197–213, 2003.

[21] Chen, X., Chugg, K. M., and Neifeld, M. A., "Near optimal parallel distributed data detection for page-oriented optical memories," *IEEE J. Select. Top. Quantum Electron.*, vol. 4, pp. 866–879, 1998.

[22] Chou, W. and Neifeld, M. A., "Interleaving and error correction in volume holographic memory systems," *Appl. Opt.*, vol. 37, pp. 6951–6968, 1998.

[23] Chou, W. and Neifeld, M. A., "Soft-decision array decoding for volume holographic memory systems," *J. Opt. Soc. Am. A*, vol. 18, pp. 185–194, 2001.

[24] Chung, S. Y., *On the construction of some capacity-approaching coding schemes*. PhD thesis, Massachusetts Institute of Technology, 2000.

[25] Chung, S. Y., J.Richardson, T., and Urbanke, R. L., "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, 2001.

[26] Cormen, T. H., Leiserson, C. E., and R. L. Rivest, C. S., *Introduction to Algorithms*. The MIT Press; 2nd edition, 2001.

[27] Di, C., Proietti, D., Telatar, I. E., Richardson, T., and Urbanke, R., "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570 –1579, 2002.

[28] Di, C., Richardson, T., and Urbanke, R., "Weight distributions: How deviant can you be?," in *2001 IEEE International Symposium on Information Theory, Washington, DC*, 2001.

[29] Dousse, O., Bacelli, F., and Thiran, P., "Impact of interferences on connectivity in ad hoc networks," *IEEE/ACM Trans. Networking*, vol. 13, pp. 425–436, 2005.

[30] Dousse, O. and Thiran, P., "Connectivity vs capacity in dense ad hoc networks," *IEEE Infocom, New York, NY, USA*, 2004.

[31] Dousse, O., Thiran, P., and Hasler, M., "Connectivity in ad-hoc and hybrid networks," *IEEE Infocom, New York, NY, USA*, 2002.

[32] Dubhashi, D., Hggstrm, O., and Panconesi, A., "Connectivity properties of bluetooth wireless networks," submitted.

[33] Eschenauer, L. and Gligor, V. D., "A key management scheme for distributed sensor networks," in *the 9th ACM conference on computer and communication security*, pp. 41–47, November 2002.

[34] Etesami, O., Molkarae, M., and Shokrollahi, A., "Raptor codes on symmetric channels," preprint, 2002.

[35] F. Mok, G. B. and Psaltis, D., "System metric for holographic memory systems," *Opt. Lett.*, vol. 21, pp. 896–898, 1996.

[36] Forney, G. D., Koetter, R., Kschischang, F., and Reznik, A., *Codes, Systems, and Graphical Models*, ch. On effective weights of pseudocodewords for codes defined on graphs with cycles, pp. 101–112. Springer, 2001.

[37] Frey, B. J., Koetter, R., and Vardy, A., "Signal-space characterization of iterative decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 766–781, 2001.

[38] Galleger, R. G., *Low-density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[39] Ganesan, D., Govindan, R., Shenker, S., and Estrin, D., "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review (MC2R '02)*, 2002.

[40] Garey, M. R. and Johnson, D. S., *Computers and Intractability, A guide to the theory of NP-Completeness.* W.H. Freeman and company, 1978.

[41] Grossglauser, M. and Tse, D., "Mobility increases the capacity of ad-hoc wireless networks," *IEEE Infocom, Anchorage, Alaska, USA*, April 2001.

[42] Gupta, P. and Kumar, P. R., "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[43] Gupta, P. and Kumar, P. R., "Towards an information theory of large networks: an achievable rate region," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1877–1894, 2003.

[44] Gupta, P. and Kumar, P., "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin and Q. Zhang (Eds.)*, 1998.

[45] Ha, J. and McLaughlin, S., "Optimal puncturing distributions for rate-compatible low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2824–2836.

[46] Ha, J. and McLaughlin, S., "Optimal puncturing of low-density parity-check codes," *In Proc. IEEE ICC*, 2003.

[47] Haas, Z., Halpern, J., and Li, L., "Gossip-based ad hoc routing," *IEEE Infocom, New York, NY, USA*, 2002.

[48] Hagenauer, J., "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. comm.*, vol. 36, no. 4, pp. 389–400, 1988.

[49] Hagenauer, J., "Rate-compatible punctured turbo (RCPT) codes in a hybrid fec/arq system," *in Proc. communications theory Mini-Conf. of IEEE GLOBECOM '97*, pp. 55–59, 1997.

[50] Heanue, J. F., Bashaw, M. C., and Hesselink, L., "Volume holographic storage and retrieval of digital data," *Science*, vol. 265, pp. 749–752, 1994.

[51] Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H., "An application specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.

[52] Intanagonwiwat, C., Govindan, R., and Estrin, D., "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Mobile Computing and Networking*, pp. 56–67, 2000.

[53] Janson, S., Lucszak, T., and Rucinski, A., *Random Graphs.* John Wiley and Sons, Inc., 2000.

[54] J.Richardson, T., Shokrollahi, M. A., and Urbanke, R. L., "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, 2001.

[55] Jung, P. and Plechinger, J., "Performance of rate-compatible punctured turbob codes for mobile radio applications," *IEEE Electronics Letters*, vol. 33, pp. 389–400, 1997.

[56] Justesen, J., "Coding for frames," in Proc. of Fortieth Annual Allerton Conference, Urbana-Champaign, IL, Oct. 2002.

[57] Karp, B. and Kung, H. T., "GPSR: gready primeter stateless routing for wireless sensor networks," in *The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom '00)*, 2000.

[58] Karp, B. and Hung, H. T., "Gpsr: Greedy perimenter stateless routing for wireless network," *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[59] Kavčić, A., Ma, X., and Mitzenmacher, M., "Binary inersymbol interference channels: Gallager codes, density evolution and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1636–1652, 2003.

[60] Khandekar, A., *Graph-based Codes and Iterative Decoding.* PhD thesis, California Institute of Technology, 2002.

[61] Kilgus, C. C. and Gore, W. C., "A class of cyclic unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 687–690, 1972.

[62] Koetter, R. and Vontobel, P. O., "Graph-covers and iterative decoding of finite length codes," in Proc. of 3rd International Symposium on Turbo Codes and Related Topics, Brest, France, Sept. 2003.

[63] Kou, Y., Lin, S., and Fossorier, M. P. C., "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2711–2736, 2001.

[64] Li, J., Blake, C., De Couto, D. S. J., Lee, H. I., and Morris, R., "Capacity of ad hoc wireless networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, (Rome, Italy), pp. 61–69, July 2001.

[65] Li, X. Y., Wan, P., Wang, Y., and Yi, C. W., "Fault tolerant deployment and topology control in wireless networks," *ACM Symposium on Mobile Ad Hoc Networking and Computing, MOBIHOC 2003.*

[66] Litsyn, S. and Shevelev, V., "On ensembles of low-density parity-check codes: asymptotic distance distributions," *IEEE Trans. Inform. Theory*, vol. 48, pp. 887–908, 2002.

[67] Litsyn, S. L. and Shevelev, V. S., "Distance distributions in ensembles of irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 3140–31595, 2003.

[68] Liu, B., Liu, Z., and Towsley, D., "On the capacity of hybrid wireless networks," *IEEE Infocom, San Francisco, CA, USA*, 2003.

[69] Luby, M., "LT-Codes," in *43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.

[70] Luby, M., Mitzenmacher, M., Shokrollahi, M., and Spielman, D., "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569–584, 2001.

[71] Luby, M., Mitzenmacher, M., Shokrollahi, M., and Spielman, D., "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585–598, 2001.

[72] MacKay, D. J. C., "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, 1999.

[73] Mandelbaum, D., "Unequal-error-protection codes dervied from difference sets," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 686–687, 1967.

[74] Manjeshwar, A. and Agrawal, D. P., "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002.

[75] Masnick, B. and Wolf, J., "On linear unequal error protection codes," *IEEE Trans. Inform. Theory*, vol. IT-3, pp. 600–607, 1967.

[76] McMichael, I., Christian, W., Pletcher, D., Chang, T. Y., and Hong, J. H., "Compact holographic storage demonstrator with rapid access," *Appl. Opt.*, vol. 35, pp. 2375–2379, 1996.

[77] Meester, R. and Roy, R., *Continuum Percolation*. Cambridge University Press, cambridge UK, 1996.

[78] Miller, G. and Burshtein, D., "Bounds on the maximum likelihood decoding error probability of low density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2696–2710, 2001.

[79] Mok, F., "Angle-multiplexed storage of 5000 holograms in lithium-niobate," *Opt. Lett.*, vol. 18, pp. 915–917, 1993.

[80] Nakamura, K., Kabashima, Y., Saad, D., and Zaragoza, R. M., "Statistical mechanics of broadcast channels using low-density parity-check codes," in *2003 IEEE International Symposium on Information Theory, Yokohama, Japan*, 2003.

[81] Nasipuri, A. and Das, S. R., "On-demand multipath routing for mobile ad hoc networks," *Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN'99)*, 1999.

[82] Neifeld, M. A. and Chou, W., "Information theoretic limits to the capacity of volume holographic optical memory," *Appl. Opt.*, vol. 36, pp. 514–517, 1997.

[83] Orlitskey, A., Viswanathan, K., and Zhang, J., "Stopping set distribution of LDPC code ensembles," submitted to IEEE Trans. Inform. Theory, 2003.

[84] Oswald, P. and Shokrollahi, M. A., "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 3017–3028, 2002.

[85] Palanki, R. and Yedidia, J. S., "Rateless codes on noisy channels," in *2004 IEEE International Symposium on Information Theory, Chicago, IL*, 2004.

[86] Penrose, M., *Random Geometric Graphs*. Oxford University Press, 2003.

[87] Penrose, M. D., "On the spread-out limit for bond and continuum percolation," *Annals of Applied Probability*, vol. 3, no. 1, pp. 253–276, 1993.

[88] Penrose, M. D., "The longest edge of the random minimal spanning tree," *The Annals of Applied Probability*, vol. 6, pp. 340–361, 1997.

[89] Penrose, M. D., "On k-connectivity for a geometric random graph," *Random Structures and Algorithms*, no. 15, pp. 145–164, 1999.

[90] Penrose, M. D., "A strong law for the longest edge of the random minimal spanning tree," *The Annals of Applied Probability*, vol. 27, pp. 246–260, 1999.

[91] Penrose, M. D. and Pisztora, A., "Large deviations for discrete and continous percolation," *Advances in Applied Probability*, no. 28, pp. 29–52, 1996.

[92] Perevalov, E. and Blum, R., "Delay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategydelay limited capacity of ad hoc networks: Asymptotically optimal transmission and relaying strategy," *IEEE Infocom, San Francisco, CA, USA*, 2003.

[93] Pishro-Nik, H., Chan, K., and Fekri, F., "On connectivity properties of large-scale sensor networks," *First Annual IEEE International Conference on Sensor and Ad Hoc Communications and Networks*.

[94] Pishro-Nik, H. and Fekri, F., "Improved decoding algorithms for low-density parity-check codes," in Proc. of 3rd International Symposium on Turbo Codes and Related Topics, Brest, France, Sept. 2003.

[95] Pishro-Nik, H. and Fekri, F., "Improved decoding algorithms for low-density parity-check codes over the binary erasure channel," accepted in 37th Annual Conference on Information Sciences and Systems.

[96] Pishro-Nik, H. and Fekri, F., "On graphs of LDPC codes," in Proc. of Fortieth Annual Allerton Conference, Urbana-Champaign, IL, Oct. 2004.

[97] Pishro-Nik, H. and Fekri, F., "On LDPC codes over the erasure channel," in Proc. of Fortieth Annual Allerton Conference, Urbana-Champaign, IL, Oct. 2003.

[98] Pishro-Nik, H. and Fekri, F., "On the maximum-likelihood decoding of low-density parity-check codes on the binary erasure channel," in Proc. of in 37th Annual Conference on Information Sciences and Systems, Baltimore, MD , March 2003.

[99] Pishro-Nik, H. and Fekri, F., "Performance of low-density parity-check codes with linear minimum distance," IEEE Trans. Inform. Theory, to appear.

[100] Pishro-Nik, H. and Fekri, F., "Results on punctured low-density parity-check codes and improved iterative decoding techniques," submitted to IEEE Trans. Inform. Theory, 2003.

[101] Pishro-Nik, H. and Fekri, F., "On decoding of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 50, pp. 439–454, 2004.

[102] Pishro-Nik, H. and Fekri, F., "Performance of low-density parity-check codes with linear minimum distance," in *2004 IEEE International Symposium on Information Theory, Chicago, IL*, 2004.

[103] Pishro-Nik, H. and Fekri, F., "Results on punctured ldpc codes," in *IEEE Information Theory Workshop*, October 2004.

[104] Pishro-Nik, H., Rahnavard, N., and Fekri, F., "Nonuniform error correction using low-density parity check codes," in Proc. of Fortieth Annual Allerton Conference, Urbana-Champaign, IL, Oct. 2002.

[105] Pishro-Nik, H., Rahnavard, N., and Fekri, F., "Results on non-uniform error correction using low-density parity-check codes," in *Global Telecommunications Conference, GLOBECOM '03*, 2003.

[106] Pishro-Nik, H., Rahnavard, N., and Fekri, F., "Non-uniform error correction using low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 51, 2005.

[107] Pishro-Nik, H., Rahnavard, N., Ha, J., Fekri, F., and Adibi, A., "Low-density parity-check codes for volume holographic memory systems," *Appl. Opt.*, vol. 42, pp. 861–870, 2003.

[108] Psaltis, D. and Mok, F., "Holographic memories," *Sci. Am.*, vol. 273, pp. 70–76, 1995.

[109] Richardson, T., Shokrollahi, A., and Urbanke, R., "Finite-length analysis of low-density parity-check ensembles for the binary erasure channel," in *2002 IEEE International Symposium on Information Theory, Lausanne, Switzerland*, 2002.

[110] Richardson, T. J. in Proc. of Fortieth Annual Allerton Conference, Urbana-Champaign, IL, Oct. 2003.

[111] Richardson, T. J. and Urbanke, R. L., "Finite-length density evolution and the distribution of the number of iterations for the binary erasure channel," Available at http://lthcwww.epfl.ch/research/ldpcopt/.

[112] Richardson, T. J. and Urbanke, R. L., "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, 2001.

[113] Richardson, T. J. and Urbanke, R. L., "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 638–656, 2001.

[114] Sason, I. and Urbanke, R., "Parity-check density versus perfprmance of binary linear block codes over memoryless symmetric channels," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1611–1635, 2003.

[115] Shakkottai, S., Srikant, R., and Shroff, N., "Unreliable sensor grids: Coverage, connectivity and diameter," *In the proceedings of IEEE INFOCOM'03, San Francisco, CA, April 2003.*

[116] Shelby, R. M., Hoffnagle, J. A., Burr, G. W., Jefferson, C. M., Bernal, M.-P., Coufal, H., Grygier, R. K., Unther, H. G., Macfarlane, R. M., and Sincerbox, G. T., "Pixel-matched holographic data storage with megabit pages," *Opt. Lett.*, vol. 22, pp. 1509–1511, 1997.

[117] Shokrollahi, A., "Raptor codes," preprint, 2002.

[118] Shokrollahi, M. A., "New sequences of linear time erasure codes approaching the channel capacity," *AAECC*, pp. 65–76, 1999.

[119] Shokrollahi, M. A., "Capacity-achieving sequences," *in IMA Volumes in Mathematics and its Applications*, vol. 123, pp. 153–166, 2000.

[120] Shokrollahi, M. A., *Codes, Systems, and Graphical Models*, ch. Capacity-achieving sequences, pp. 153–166. Springer, 2001.

[121] Tanner, R. M., "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, 1981.

[122] Tian, T., Jones, C., Villasenor, J. D., and Wesel, R. D., "Construction of irregular LDPC codes with low error floor," *In Proc. IEEE ICC*, 2003.

[123] Vardy, A., "The intractability of computing the minimum distance of a code," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1757–1766, 1997.

[124] Wan, P. and Yi, C. W., "Asymptotic critical transmission radius and critical neighbor number for k-connectivity in wireless ad hoc networks," *ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2004*.

[125] Xue, F. and Kumar, P. R., "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.